OPTIMIZATION OF A QUANTUM CASCADE LASER OPERATING IN THE TERAHERTZ FREQUENCY RANGE USING A MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

THESIS

Traci A. Keller, First Lieutenant, USAF

AFIT/GCE/ENG/04-03

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GCE/ENG/04-03

Optimization of a Quantum Cascade Laser Operating in the Terahertz Frequency Range Using A Multiobjective Evolutionary Algorithm

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Traci A. Keller, B.S.

First Lieutenant, USAF

June, 2004

AFIT/GCE/ENG/04-03

Optimization of a Quantum Cascade Laser Operating in the Terahertz Frequency Range Using A Multiobjective Evolutionary Algorithm

Traci A. Keller, B.S.
First Lieutenant, USAF

Approved:

/signed/	8 Jun 2004
Dr. Gary B. Lamont Chairman	Date
/signed/	8 Jun 2004
Dr. Gilbert L. Peterson Member	Date
/signed/	8 Jun 2004
Dr. James A. Lott Member	Date

Acknowledgements

It has taken almost two years to reach the end of this endeavor. It is an accomplishment I certainly did not create on my own. First, and foremost, I would like to thank God for the many blessings He has bestowed upon me over this lifetime.

Beyond that, I would like to say thank you to my advisor, Dr. Gary B. Lamont. You pushed me to acquire knowledge outside of my area of expertise and showed me that with a lot of hard work and a little less sleep what appears an impossible goal is actually within reach. I would like to extend my gratitude to Dr. Gilbert L. Peterson for taking the time to sit on my committee and proofread my document. I would also like to thank Dr. James A. Lott for patiently explaining his area of expertise to someone who doesn't know the first thing about physics. Your knowledge helped make this document much more readable.

In addition, I would like to thank my classmates for always inventing some new form of procrastination. You kept the mood light and made the academic environment much less stressful.

To my closest friends, those I have known anywhere from four to twenty-one years, you have helped me keep my focus, offered motivation when I was lacking, been an outlet for my frustrations, listened to my ramblings and loved me every minute here.

To my sister I would like to say thank you. We have chosen very different paths in life, both filled with success defined by different measures, but your strength and perseverance have encouraged me over the past two years. I would love and cherish your presence in my life, sister, or not.

And now, last, but most importantly I must thank the two people in this world who have stood by me from the very beginning. Mom and Dad, collectively you have shown me that success in this world is much more than money and power. Separately, Dad, you are my lighthouse. Your character, work ethic, values and dedication are my source of strength and my estimation of right from wrong. Mom, you have always been my greatest supporter, always an incredible mother to me, but lately, and now more importantly, you are my friend. Thank you for continually lending a nonjudgmental ear to my trials and heartaches. My success in this lifetime is a direct reflection of the amazing people you two

are and have always been. Know that I love you and how blessed I feel to be your 'baby girl'

Traci A. Keller

Table of Contents

Page

Page

List of Figures

List of Tables

List of Abbreviations

Abbreviation		Page
(QC)	Quantum Cascade	2
(AOI)	Applied Optoelectronics, Inc.	3
(THz)	Terahertz	3
(MOP)	Multiobjective Optimization Problem	7
(EA)	Evolutionary Algorithm	8
(GA)	Genetic Algorithm	8
(MOEA)	Multiobjective Evolutionary Algorithm	8
(GenMOP)	General Mutliobjective Parallel Genetic Algorithm	8
(QW)	Quantum Well	15
(SQW)	Single Quantum Well	15
(QSE)	Quantum Size Effects	16
(SL)	Superlattice	17
(MBE)	Molecular Beam Epitaxy	17
(CAR)	Continual Azimuthal Rotation	20
(RHEED)	Reflection High-energy Electron Diffraction	20
(PF^*)	Pareto Front	24
(DM)	Decision Maker	27
(GPGA)	Global Parallel Genetic Algorithm	30
(mgGA)	Micro-grain Genetic Algorithm	30
(cgGA)	Coarse-grained Genetic Algorihtm	30
(DGA)	Distributed Genetic Algorithm	30
(fgGA)	Fine-grained Genetic Algorithm	32
(MPGA)	Massively Parallel Genetic Algorithm	32
(GaAs)	Gallium Arsenide	37
(FEM)	Finite Element Method	39

AFIT/GCE/ENG/04-03

Abstract

A quantum cascade (QC) laser is a specific type of semiconductor laser that operates through principles of quantum mechanics. In less than a decade QC lasers are already able to outperform previously designed double heterostructure semiconductor lasers. Because there is a genuine lack of compact and coherent devices which can operate in the far-infrared region the motivation exists for designing a terahertz QC laser. A device operating at this frequency is expected to be more efficient and cost effective than currently existing devices. It has potential applications in the fields of spectroscopy, astronomy, medicine and free-space communication as well as applications to near-space radar and chemical/biological detection.

The overarching goal of this research was to find QC laser parameter combinations which can be used to fabricate viable structures. To ensure operation in the THz region the device must conform to the extremely small energy level spacing range from 10-15 meV. The time and expense of the design and production process is prohibitive, so an alternative to fabrication was necessary. To accomplish this goal a model of a QC laser, developed at Worchester Polytechnic Institute with sponsorship from the Air Force Research Laboratory Sensors Directorate, and the General Multiobjective Parallel Genetic Algorithm (GenMOP), developed at the Air Force Institute of Technology, were integrated to form a computer simulation which stochastically searches for feasible solutions.

GenMOP is a pareto-based algorithm that utilizes real values for crossover and mutation operators. Additionally, the algorithm employs fitness sharing through a niche radius. The individual chromosomes are encoded with real-values denoting the temperature, bias, current density, layer thickness and donor density of a particular laser. Auxiliary genes are associated with the individual chromosomes to define fitness values and pareto ranking.

The GA investigates 17 distinct frequencies, ranging from 1.0 to 6.0 THz, at 3 separate temperatures in search of feasible answers. Great difficulty is created for the GA in its pursuit of feasible solutions due to the dimensionality and complexity of the search space, as well as the relative scarce existence of these solutions.

Optimization of a Quantum Cascade Laser Operating in the Terahertz Frequency Range Using A Multiobjective Evolutionary Algorithm

1. Introduction

Half of the 2000 Nobel Prize in Physics was awarded to Zhores Alferov and Herbert Kroemer for their participation in the development of the double heterostructure, or semiconductor laser. Jack Kilby, of Texas Instruments, Inc., was awarded the other half for his invention of the monolithic integrated circuit. The semiconductor lasers developed by Alferov and Kroemer have had an incredible impact on our lives. Because of their small size, ease of use and reliability they are found in such everyday devices as laser pointers, compact disc players, laser printers and fax machines. Semiconductor lasers are composed of an active p-n junction diode region positioned between two waveguide cladding layers that provide for the movement and recombination of electrons and holes. With an applied voltage these cladding layers provide either the electrons or the holes for movement through the energy bandgap that exists between the conduction and valence bands. The energy bandgap of the active region of the semiconductor device primarily determines the wavelength of the light emitted. Semiconductor lasers of this type have long been available in the commercial market, but their usefulness significantly diminishes at wavelengths greater than $\sim$*1.6* μm. At peak emission wavelengths greater than *2* μm semiconductor laser materials become sensitive to the repetitious cooling and heating that occurs during laser operation [12]. This property makes double heterostructure laser diodes inefficient and unreliable devices at a wavelength greater than *2* μm, especially when operated at, or above room temperature.

This research attempts to create a computer simulation through the integration of a QC laser model, developed at Worchester Polytechnic Institute, and the General Multiobjective Parallel (GenMOP) GA, developed at the Air Force Institute of technology. It is hoped that this GA originally designed for solution to a groundwater remediation problem [30] can produce combinations of QC laser parameters which can be fabricated into viable

semiconductor heterostructures. The following section gives an historical prospective of the problem while Section 1.2 describes the motivation for the research, Section 1.3 outlines the objectives and Section 1.4 gives an overview of the remainder of the document.

1.1 Quantum Cascade Laser Background

The mid-infrared region of the electromagnetic spectrum, *2 - 20* μm was of great interest to scientists, so additional devices possessing the ability to lase within this region needed to be developed. From the Ioffe Physico-Technical Instistute in St. Petersburg, Russia came the basic concept behind quantum cascade lasers. In 1971 two Russian scientists, Rudolf Kazarinov and Robert Suris suggested that optical amplification could occur *between* energy levels in a quantum well structure if a bias voltage was applied. Their proposal for intersubband radiative recombination was the basic concept behind today's quantum cascade (QC) lasers. Unfortunately, their proposition required procedures for the repetitive growth of epitaxial crystal layers with thicknesses on the order of tens of monolayers. The precision required for heterostructure growth was not available until the late 1980s. The evolution of an epitaxial growth technique, known as molecular beam epitaxy (MBE), through the 1970s and 80s eventually made the demonstration of the QC laser possible. The precise crystal composition and doping control made feasible by the application of MBE to the semiconductor heterojunction device design process is now called bangap engineering [9]. It would take a considerable effort and more than two decades before the quantum cascade laser would first be demonstrated by a research group led by Federico Capasso and Alfred Cho at Bell Laboratories in 1994 [12].

A quantum cascade laser is a specific type of semiconductor laser that operates through principles of quantum mechanics. The unipolarity of a QC laser indicates that electrons are solely responsible for releasing energy in the form of photons. These electrons transition from one quantum energy state to another within a layer, or group of layers, of semiconductor material releasing energy in the form of photons during their descent. The binding energy necessary to pull these electrons away from the Coulombic force of the nucleus in the atom is related to the extremely thin semiconductor layers. A property of quantum mechanics known as quantum confinement occurs when the electrons are trapped within a thin semiconductor quantum well layer. These electrons can freely move in only

two directions within the plane of the thin layer. In this case quantum confinement, which leads to discrete energy levels that electrons can occupy in a material smaller than the de Broglie wavelenth, $\sim 30nm$, occurs in only one dimension due to the the quantum well structure [63]. Unlike the earliest form of semiconductor lasers where the energy bandgap determines the wavelength of the light emitted, with QC lasers the thickness of the layers determines the wavelength. This is a critically important property of QC lasers because it allows them to be tuned to a desired frequency through bandgap engineering.

In less than a decade QC lasers are already able to outperform previously designed double heterostructure semiconductor lasers. Quantum cascade lasers can be tuned to operate over the entire mid-infrared range and even into the far-infrared range of the electromagnetic spectrum. They are able to generate hundreds of milliwatts of power in pulsed operation at room temperature. Continuous wave operation has also been reported at room temperature by Jerome Faist's group from the University of Neuchatel in Switzerland. This group achieved continuous wave operation at an emission wavelength of 9.1 μm with 17 mW of peak output power using a four quantum well structure design for their active region [77]. The device was made of an InGaAs/InAlAs heterostructure with a four quantum well structure that is repeated thirty-five times. Using techniques which minimized the threshold current density and laser geometry for heat dissipation the device performed continuous wave operation at room temperature (292 K) [7]. Commercially available QC lasers are manufactured by Applied Optoelectronics, Inc. (AOI). Quantum cascade lasers are also commercially available from Alpes Lasers, with Physical Sciences, Inc., including products utilizing QC lasers for combustion diagnostics and environmental sensing [13].

1.2 Motivation

Because there is a genuine lack of compact and coherent devices which operate in the far-infrared area of the electromagnetic spectrum, the motivation exists for designing a QC laser that operates from 1.0 to 6.0 terahertz (THz). Additionally, the QC laser is expected to be more efficient and cost effective than currently existing devices operating in the terahertz frequency range. It has potential applications in the fields of spectroscopy, astronomy,

medicine and free-space communication, as well as applications in near-space radar systems and in chemical/biological detection due to the unique signatures of biomolecules in the terahertz frequency range [68].

Within the medical community it is hoped that terahertz radiation will lead to more detailed pictures of soft tissue than ultrasound and x-ray images are now able to produce. This would furnish medical providers with the ability to observe the healing progress of a wound without removing the bandage, the severity of a skin burn, or the presence of a tooth cavity. In addition, skin and breast cancer could be detected through detailed images and the capability to scan below the epidermis. All of these capabilities are available with terahertz radiation because the low photon energies eliminate the risk of photoionization in tissue [67].

The terahertz region of the electromagnetic spectrum is also believed to be the real "fingerprint" region for chemical and biological signatures, but it is the least characterized. This range in the electromagnetic spectrum does not experience a large amount of water vapor absorption promoting a transparency which allows detection of chemical and biological agents through spectroscopic methods [77]. Additionally, terahertz frequency emission has the ability to non-invasively permeate paper, ceramic and cardboard membranes. Single, or multiple detectors with a reasonable level of power producing an acceptable signal-to-noise ratio may be tuned for detection of a specific agent. In addition to chemical and biological detection terahertz emission has the ability to detect a concealed weapon using 2-D fast Fourier transforms to produce an image array [68].

QC lasers may be utilized in many real-world applications ranging from communication to chemical/biological agent detection. The combination of bandgap engineering and molecular beam epitaxy (MBE) for device fabrication are capable of producing a semiconductor device which operates in the far-infrared region of the electromagnetic spectrum [77]. Through manipulation of QC laser parameters such as applied bias and semiconductor layer thickness optimization for a QC laser operating at a terahertz frequency can be achieved.

1.3 Research Goals and Approach

The overarching goal of this research is to determine parameter combinations for a quantum cascade laser which operates continuously in the terahertz frequency range. To accomplish this goal, four main objectives must be completed. Each of these goals is an important step in realizing a solution to this multiobjective problem of QC laser optimization.

Objective 1. Describe the Quantum Cascade Laser problem domain.

Objective 2. Analyze methods for solving multiobjective optimization problems.

Objective 3. Define the specific algorithm used to determine solution(s).

Objective 4. Analyze the results to determine a direction for future research.

Possessing a clear understanding of the theory behind lasers is essential because complications arise when attempting to comprehend the intricacies in the development of a laser with a specified purpose. Each detail in the progression maintains its own impact on the overall performance of the final laser. It is imperative that each aspect of the problem domain be fully understood.

Quantum cascade lasers are part of the family of semiconductor lasers. Although well documented and understood within published literature they are fairly new in their development especially those devices operating in the terahertz frequency range. It is valuable to first understand the basic concepts behind lasers and their construction before defining QC lasers as stated in Objective 1 above. Steps 1 to 5, below, outline the description the QC laser problem domain is given in subsequent chapters.

Step 1.1 Describe the theory behind lasers.

Step 1.2 Define semiconductors and their construction for creating a laser.

Step 1.3 Illustrate the principles behind quantum cascade lasers.

Step 1.4 Define preferred performance characteristics for lasers both in general and for a specific laser.

Step 1.5 Describe creation of specific semiconductor used to satiate the goals in 1.4.

Generally, laser optimization requires the tuning of parameters to improve the efficiency, increase the output power, change the emission wavelength, or mode of operation, improve the beam quality, or increase the overall gain. In this case, the optimization of the quantum cascade laser is a multiobjective problem because in addition to being concerned about the gain it is also necessary to ensure the laser is operating in the terahertz frequency range. In order to fully understand multiobjective optimization and the methods generally used to obtain solutions the following steps are followed:

Step 2.1 Define a generic multiobjective optimization problem.

Step 2.2 Determine common methods used for solutions.

2.2.a Describe simulated annealing.

2.2.b Describe Tabu search.

2.2.c Describe evolutionary algorithms and specific areas.

2.2.c.1 Define artificial immune systems.

2.2.c.2 Define genetic algorithms.

Step 2.3 Reference specific real-world problems within each description.

After a clear understanding of methods for multiobjective optimization is achieved the method deemed most appropriate for discovering solutions to the QC laser problem is chosen. An algorithm is designed which takes into account the particular complexities and requirements of the problem at hand. This method is determined, described, implemented and tested through the steps outlined below:

Step 3.1 Determine the best method for solution to QC laser optimization problem.

Step 3.2 In detail, describe the algorithm.

Step 3.3 Implement the algorithm.

Step 3.4 Design a set of experiments to create viable solution(s).

The final goal in this research is to analyze the results achieved through simulation of the quantum cascade laser. The effectiveness of the algorithm will be determined in response to the number and type of solutions found. This is an important step in the

process because it gives insight into the completeness of the research. Additionally, the effectiveness of the algorithm helps determine the direction of future research. Should the algorithm be deemed successful in finding solutions, i.e. deemed effective, then additional constraints may be added and other alterations to the problem domain may be made while continuing to utilize the developed algorithm. If the algorithm is determined to be ineffective, then it may become necessary to either modify the existing algorithm, or design an entirely new algorithm more suited to what has been learned about the problem domain. These options are discussed in greater detail in the future work section of Chapter 8. In addition to the effectiveness of the algorithm the efficiency is also measured. This measurement is determined through two separate calculations. The first is a measure of the speedup of the parallel algorithm compared to its serial counterpart. Also, a determination is made about the efficiency of the algorithm by comparing the number of solutions found to the time taken for algorithm operation. The process described is reflected in the following steps:

Step 4.1 Determine the effectiveness of the chosen algorithm.

Step 4.2 Measure the efficiency of chosen the algorithm.

Step 4.3 From the efficiency and effectiveness metrics determine a direction for future research.

1.4 Thesis Overview

Understanding the problem domain is the first priority in this research. Chapter 2 provides the background knowledge necessary to appreciate the quantum cascade laser structure and operation. Beginning with the basic structure of an atom, following through laser fundamentals, semiconductors, quantum wells and finally ending with a discussion of generic QC lasers.

With the *a priori* knowledge that optimization of a QC laser is a multiobjective optimization problem (MOP) Chapter 3 begins with a description of this problem type. Following this is a discussion of some algorithms found in literature and used for solving MOPs. An introduction and comprehensive discussion of all algorithms applied to MOPs

is outside the scope of this research, but those most commonly found are incorporated. These include simulated annealing, Tabu search, evolutionary algorithms (EA) such as the artificial immune system and genetic algorithms (GA), and multiobjective evolutionary algorithms (MOEA) . Because a MOEA that has parallel capabilities was selected for solving the multiobjective QC laser optimization problem this chapter concludes with a discussion of parallelization techniques for GAs.

Further details of the problem domain are given in Chapter 4. A deeper discussion into the intricacies of QC laser operation is offered. The particular material chosen for modeling is described followed by the mathematics necessary to create an accurate software model.

The specific algorithm designed for use with the QC laser software model is defined in Chapter 5. The implementation details for the concepts of crossover, mutation and equivalence class sharing utilized in the General Multiobjective Parallel Genetic Algorithm (GenMOP) are discussed.

Chapter 6 discusses the efficiency and effectiveness goals of this research. It outlines the experiments designed towards these ends and the measurements taken.

The results of the experiments are presented in Chapter 7. The efficiency is discussed with reference to the speedup achieved, efficiency as defined in [52] and the speedup achieved for a master/slave parallel GA model. The effectiveness of the algorithm is shown using the Kruskal-Wallis probability test and demonstrating the exploration of the search space using GenMOP.

The research is concluded in Chapter 8 and directions for future work are discussed.

2. Quantum Cascade Laser Problem Domain

In 1808 John Dalton, an English teacher and scientist, proposed what has come to be known as modern atomic theory. Building upon Democritus' idea of the atom from 530 B.C. Dalton suggested that every element in the universe was made of atoms which were able to combine and form compounds. He also believed that all atoms of an element were identical, but each specific element's physical properties differed from those of all other elements. Just as everything else in the universe, lasers are composed of the most basic element that is able to keep its chemical properties, the atom. Because this is true it is important to understand the structure and characteristics of an atom in order to fully understand lasing.

Each section in this chapter builds upon the previous to create a foundation for understanding the structure and operation of quantum cascade lasers. Fundamental concepts such as the atom, lasers and semiconductors are introduced in the next few sections. Section 2.4 gives a brief overview of quantum well lasers followed by a discussion of generic properties of quantum cascade lasers in Section 2.5. The chapter concludes with a discussion of an epitaxial growth process, molecular beam epitaxy, used for fabrication of these heterostructures.

2.1 Atoms

The most commonly recognized model of the atom comes from the field of quantum mechanics which studies the motion of particles through their wave properties. The Bohr model takes ideas from two other areas of science in combination to correctly model the three particles every atom contains. Figure 1 represents Bohr's vision of an atom with a chart describing the difference in energy between the levels where electrons persist in orbit around the nucleus. The energies at these levels are measurable, definitive values that correspond to the energy required to remove an electron from the quantized level.

In the field of radioactivity the nucleus of an atom and its two subatomic particles were discovered. The proton, a positively charged particle has a mass of $1.673x10^{-24}grams$. In addition, the neutron, whose mass is $1.675x10^{-24}grams$, also resides in the nucleus.

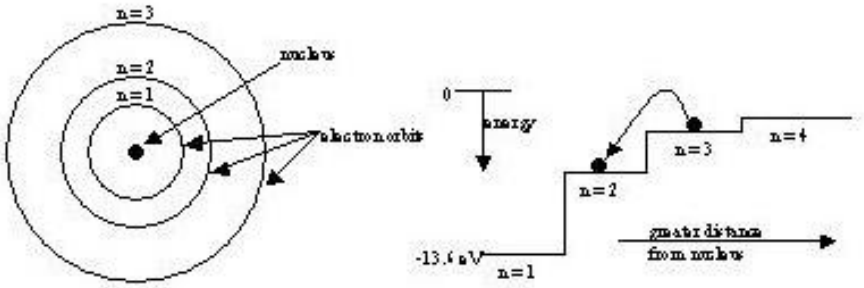

Figure 1 Bohr Atomic Model [71]

Concurrently the electron, a negatively charged particle, was discovered in the chemistry field by J. J. Thomson. Although he couldn't prove the electrons existence, through experimentation Thomson postulated that electrons were a fundamental part of the atom. With the discovery of the electrical charge by Robert Milikan physicists were able to accurately calculate that the electron weighs $9.10x10^{-28}grams$. To comprehend the operation of a laser it is most important to understand the mechanics of an electron. Each electron contained by an atom is confined to orbit in a specific shell surrounding that atom's nucleus. While the Bohr model denotes all shells as being perfectly circular they are actually less defined shapes that relate to an energy level. Using the idea of undefined orbits and Heisenberg's uncertainty principle, which describes the relationship between knowledge of a subatomic particle's position and momentum, Schrödinger derived a set of wave functions that define each electron's energy level, velocity, orientation and direction. An electron's principal quantum number denotes which energy level it resides within at a resting state. The knowledge of this energy level is valuable in understanding the lasing properties of a particular atom or compound.

2.2 Lasers

Laser, although a word commonly utilized in the English language, is actually an acronym standing for *light amplification by stimulated emission of radiation.* In essence, a laser is a device designed to control the photon emission of an atom, or group of atoms,

when it moves from an "excited level" back to its ground-state energy level. When the electron changes from a higher to a lower energy level the excess energy may be emitted in the form of a photon, or light energy. This excess energy is equal to the difference between the excited energy state where the electron began, E_2, and the energy state where it relocated, E_1. This transformation is illustrated in Figure 2.

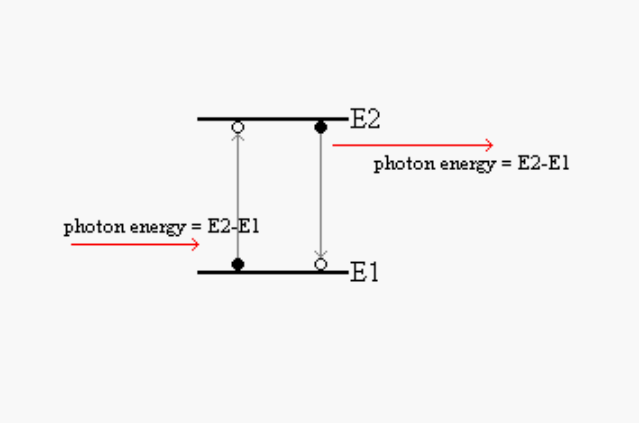

Figure 2 Photon Emission and Absorption via Electron Movement from E_2 to E_1 and E_1 to E_2 [87]

A photon's exact energy can be calculated using Equation 1 where ν is the frequency of light and h represents Planck's constant where $h = 6.63x10^{-34} J * s$.

$$Photon\ energy = h * \nu = \Delta E = E_2 - E_1 \tag{1}$$

The electron's change in energy-state at the point of photon release determines the wavelength and thus the color of the light emitted. This occurence is shown in Equation 2 where the energy in electron volts is related to 1.24 divided by the wavelength in μm.

$$E = \frac{hc}{\lambda} \approx \frac{1.24}{\lambda} \tag{2}$$

This emission of light holds true to three properties which distinctly separate it from normal light.

1) The light is monochromatic, or one particular color.

2) Light is coherent. It is organized and each photon's wavefront is in unison for stimulated emission.

3) Light is directional because it is in a tight, strong beam for stimulated emission.

These three properties hold in the model developed by Albert Einstein in 1917 for what he called stimulated emission [69]. Figure 3 shows this property which expresses an individual photon's ability to cause an atom to emit a photon that will vibrate with the same frequency and direction as the original photon.

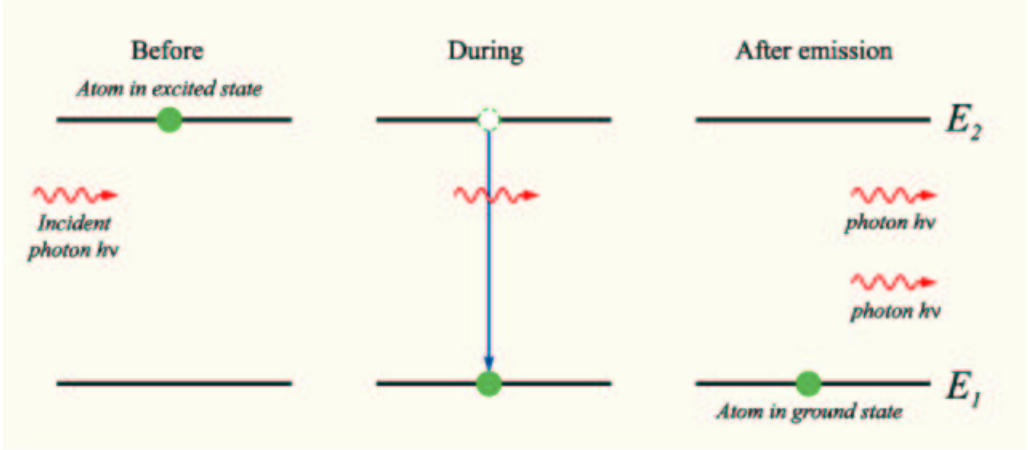

Figure 3 Stimulated Emission Process [86]

Einstein's stimulated emission property alone is not enough to ensure that lasing occurs. To overcome the effects of absorption, which essentially negates the energy released from an electron by inducing the excitation of another electron, population inversion must occur. For QC lasers the lasing device must be designed, so that electrons are able to tunnel between wells of identical energy in less time than they remain in their highest energy level. In this way population inversion guarantees that more electrons exist in their excited state than in their ground state, so that lasing is possible.

2.3 Semiconductors

Semiconductor devices are found in almost all modern day systems, such as ignition modules in automobiles and central processing units in microcomputers. These devices may be composed of many different atomic materials, but each device within the entire

semiconductor family falls into one of two major categories: elemental, or compound semiconductors. A listing of several common semiconductor compositions falling within these categories can be found in Table 1 [65].

Table 1 **Semiconductor Materials**

General Classification	Symbol	Semiconductor Name
Elemental	Si	Silicon
	Ge	Germanium
Compounds		
IV - IV	SiC	Silicon carbide
III - V	AlP	Aluminum phosphide
	AlAs	Aluminum arsenide
	GaN	Gallium nitride
	GaP	Gallium phosphide
	GaAs	Gallium arsenide
	InP	Indium phosphide
	InAs	Indium arsenide
	InSb	Indium antimonide
II - VI	ZnO	Zinc oxide
	ZnS	Zinc sulfide
	ZnSe	Zinc selenide
	CdS	Cadmium sulfide
	CdSe	Cadmium selenide
	HgS	Mercury sulfide
IV - VI	PbS	Lead sulfide
	PbSe	Lead selenide

A semiconductor is a crystalline solid where all the atoms are arranged in a highly ordered structure often called a lattice. The underlying lattice in a crystal structure facilitates a method for crystal classification. In general, there are five Bravais lattices in two-dimensional space and fourteen found in three-dimensions which can be repeated to fill a defined space. The 2-D lattices are square, rectangular, centered rectangular, hexagonal and oblique while the fourteen 2-D lattices are summarized in Table 2.

Each lattice structure is defined by three unit vectors, $\vec{a_1}$, $\vec{a_2}$, $\vec{a_3}$ with α, β and λ denoting the angles between the unit vectors. For semiconductors, the most common lattice structure seen in one variation or another is the diamond lattice shown in Figure 4 where a denotes the cube side length [84]. This diamond structure can be created from

Table 2 **Bravais Three-Dimensional Lattices**

Name	No. of Bravais Lattices	Conditions
Triclinic	1	$a_1 \neq a_2 \neq a_3$, $\alpha \neq \beta \neq \lambda$
Monoclinic	2	$a_1 \neq a_2 \neq a_3$, $\alpha = \beta = 90° \neq \lambda$
Orthorhombic	4	$a_1 \neq a_2 \neq a_3$, $\alpha = \beta = \lambda = 90°$
Tetragonal	2	$a_1 = a_2 \neq a_3$, $\alpha = \beta = \lambda = 90°$
Cubic	3	$a_1 = a_2 = a_3$, $\alpha = \beta = \lambda = 90°$
Trigonal	1	$a_1 = a_2 = a_3$, $\alpha = \beta = \lambda$ ¡ $120° \neq 90°$
Hexagonal	1	$a_1 = a_2 \neq a_3$, $\alpha = \beta = 90°$ $\lambda = 120°$

two face-centered cubic lattices whose atoms each form four covalent bonds with adjacent atoms giving the structure its diamond pattern.

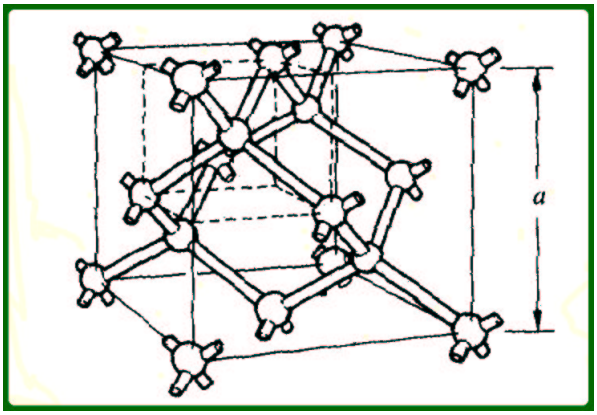

Figure 4 Diamond Lattice Unit Cell [65]

The original concept describing semiconductor lasers was developed in 1961 by Basov, et al. This group believed that the property of stimulated emission would hold when carriers were injected across a p-n junction and allowed to recombine. Semiconductor lasers were demonstrated independently by three separate laboratories the following year, but these lasers were fairly inefficient due to their high threshold currents which required operation to be at cryogenic temperatures, i.e.77 K. Due to this inefficient operation scientists were prompted to develop the heterostructure which replaces the p-n junction with layers of semiconductor material through an epitaxial growth process [16].

The epitaxial growth process of a conventional semiconductor laser diode typically begins with a n-doped semiconductor substrate. On top of this substrate a n-doped cladding layer is produced, followed by an undoped quantum well active region. The diode is finished

with a p-doped cladding layer. Figure 5 is a simplistic diagram of the generic semiconductor laser diode described above. The small amount of impurity that is mixed into the crystal structure during doping forces the device to behave as a conductor. Current is infused through the electrodes attached to the top and bottom of the structure. A voltage drop, or forward bias occurs across the p-n junction shown in Figure 5 and lasing happens [15]. Semiconductor lasers have the ability to convert input electrical energy directly into output coherent light energy. The power conversion efficiency can be as high as 90%, where edge-emitting laser diodes can generate a few Watts of output power. Although this is the case typically laser diodes are designed and operated to emit tens, or hundreds of milli-Watts of optical output power at input bias currents of tens, or hundreds of milliamperes [69].

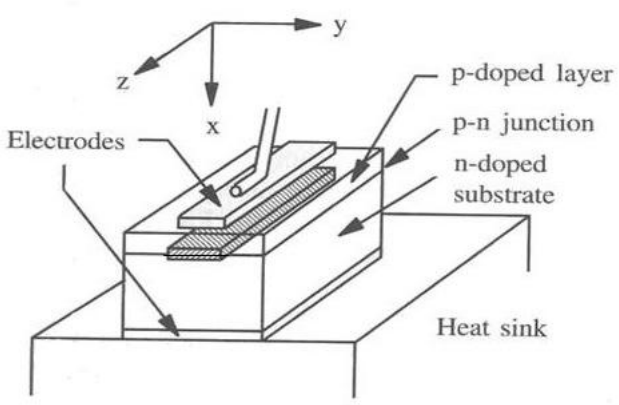

Figure 5 Semiconductor Laser with Active Region Shaded. [15]

2.4 Quantum Well Lasers

A quantum well (QW) structure is formed when a thin layer of a narrow bandgap semiconductor is positioned between two potential barriers with a higher energy bandgap. Figure 6 illustrates a single quantum well (SQW) structure where E_g denotes the bandgap energies of the two semiconductors. The bandgap energy is calculated with Equation 3 where x represents the intrinsic carrier concentration.

$$E_g = \begin{cases} 1.424 + 1.247x & \text{when x} \leq .45, \\ 1.424 + 1.087x + .438x^2 & \text{when x} \leq .43. \end{cases} \tag{3}$$

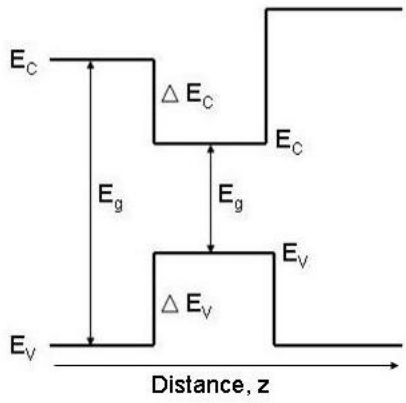

Figure 6 Single Quantum Well Semiconductor Structure.

The growth direction is shown where z represents the distance, or size of the QW. In the barrier layers electrons and holes move freely because the layer thickness exceeds the de Broglie wavelength, calculated with Equation 4 where $\hbar$ is Planck's constant and p is the momentum of the electron.

$$\lambda = \frac{\hbar}{p} \tag{4}$$

When the potential well created by the narrow bandgap semiconductor is on the order of tens of nanometers in width quantum size effects (QSE) begin to occur [28]. Quantized energy states form within these smaller potential wells and can be determined from the solution to the time independent Schrödinger wave equation in two-dimensions [59]. By varying the potential well thickness the position of quantized energy states can be manipulated [40]. This ability to control the placement of the energy states within a QW combined with the introduction of superlattice structures eventually gave way to the development of quantum cascade lasers.

2.5 Quantum Cascade Lasers

Superlattice (SL) structures, introduced by Esaki and Tsu [23] in 1970, are created through alternating layers of semiconductor materials with different bandgap energies. Building on this discovery Kazarinov and Suris proposed the possibility of lasing through a voltage applied across a SL. With the properties of tunneling injection and intersubband transitions in QWs their proposal suggested that given identical QWs through which an electron could move, a number of photons would be released. This energy emission, in the form of photons, would cause light amplification to occur within a confined QW. The physics principles supporting the proposal gave it a solid foundation, but the lack of growth techniques for semiconductor devices and the non-existence of bandgap engineering made fabrication and demonstration of such a device impossible for many years following. With the development of molecular beam epitaxy (MBE), described in detail in Section 2.6, heterostructures could be grown. As we entered into the 1980s bandgap engineering became a reality and the design of semiconductor devices with specific desired properties began. In addition to these advancements was a demonstration of resonant tunneling at Lincoln Laboratories in 1983. Population inversion could be assured within the subband of a device using the resonant tunneling technique as a pumping mechanism. When population inversion is achieved the emission dominates the absorption within the structure and lasing is possible with a quantum cascading scheme [85]. This important discovery combined with growth techniques and band-structure engineering led to the demonstration of a quantum cascade laser at Bell Labs in 1994.

Quantum cascade lasers differ from other semiconductor lasers in two major respects. First, QC lasers are unipolar because they use only electrons. A QC laser's unipolar attribute means that only one carrier is utilized for lasing. Carrier electrons make transitions within the bands of a heterostructure releasing a photon at each stage. After an electron emits a photon of a particular frequency, corresponding to the energy level it just traversed, it is then collected and injected into the next stage, so an additional photon can be emitted. Each emitter and collector/injector pair is defined as one period of the laser. This is a continuous process through the number of stages contained in the structure and

is illustrated in Figure 7. This cascading, which causes emission of photons and in turn lasing, is the attribute of the quantum cascade laser that gives it its name [55].

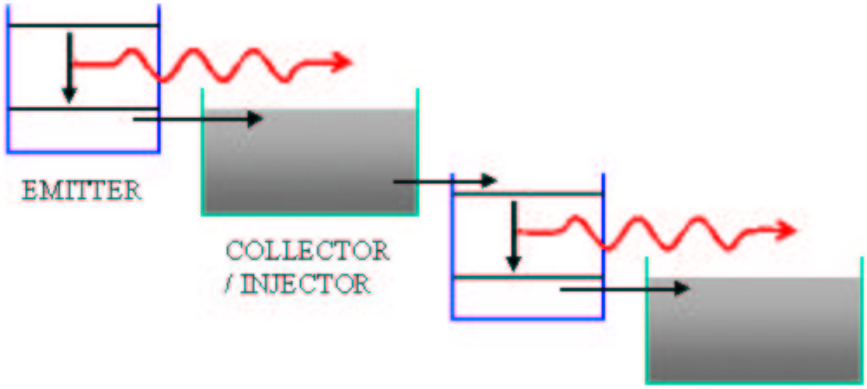

Figure 7 Cascading Scheme of a Quantum Cascade Laser [68]

The second difference between QC lasers and conventional semiconductor lasers is the cascading scheme used by QC lasers which produces recycled electrons. This recycling allows the electrons to emit photons from period to period and, in this way, contribute to the overall gain. In addition to these attributes, QC lasers are unique because their performance is not directly related to the properties of the specific semiconductor used, but rather is governed by the thickness of the fabricated layer. In essence, this means a QC laser is tunable to the terahertz frequency we are interested in here.

These devices may be fabricated using a growth process like molecular beam epitaxy to ensure a specific emission frequency by adapting the well width and barrier heights during production and the applied bias for lasing. The ability to modify the structure through its intersubband energy level spacing, discussed previously, permits the structure to lase anywhere from the mid-infrared to the far-infrared ranges of the electromagnetic spectrum. Although this straightforward explanation of QC laser devices makes the design and production process sound simple it is a daunting task. To ensure operation in the THz region the device must conform to the specific and extremely small energy level spacing range from $\sim$ 10-15 meV. Additionally, since population inversion is a necessary condition for lasing the competing non-radiative mechanisms' decay rate, described in

more detail in Chapter 4 must be overcome. Figure 8 shows the photon emission of an electron between energy levels three and two while also illustrating the non-radiative energy reduction between levels two and one where ω_{phonon} represents the interface phonon energy. The depopulation of level two must occur rapidly to ensure population inversion is maintained between levels three and two [55].

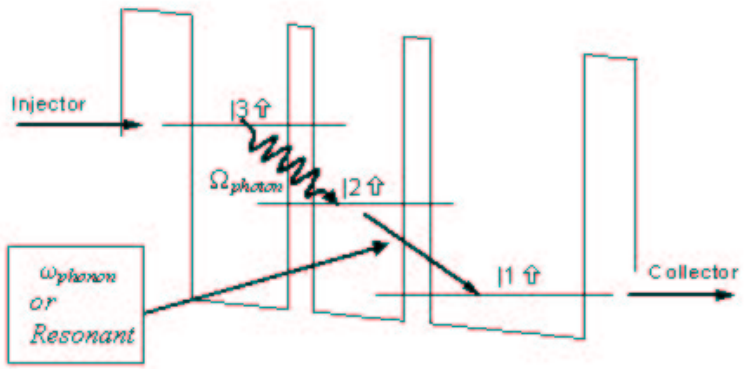

Figure 8 Three Energy Level Quantum Cascade Laser [55]

2.6 Molecular Beam Epitaxy

Quantum cascade laser heterostructures require sharp interfaces, defect and impurity free states as well as structure repeatability for satisfactory performance [55]. These inflexible requirements necessitate a growing technique with a high degree of control. Molecular beam epitaxy offers fine control over layer thickness, material composition and doping. Through the use of this technique production of high-quality structures is assured.

Molecular beam epitaxy utilizes a heated, crystalline substrate for deposition of atoms and molecules from a beam of the same epitaxial layers. The structures must be created in an ultra-high vacuum environment with pure sources to ensure the highest purity layer structures are grown. Figure 9 illustrates a generic MBE system with a growth chamber and other essential parts.

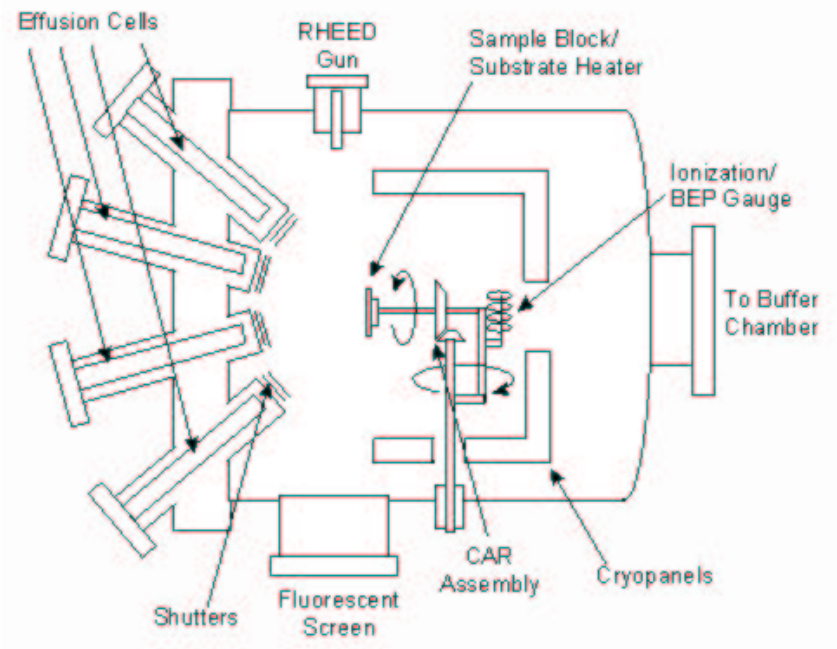

Figure 9 Molecular Beam Epitaxy Growth Chamber and System Parts. [66]

The elements required for fabrication, in this case Ga, Al and As, are housed in the effusion cells, or ovens, which use heat to control the rate of element evaporation. The shutters connected to the ends of the cells mechanically control the flow of the molecular beams. To ensure that each layer is grown uniformly the substrate is constantly rotated using the continual azimuthal rotation (CAR) assembly. The reflection high-energy electron diffraction (RHEED) gun, shown at the top of Figure 9, is used to take in-situ measurements of the structure during the growth process, which occurs in an ultra-high vacuum to create the necessary purity [54].

2.7 Summary

Quantum cascade lasers are complex structures that require background knowledge of some fundamental physics concepts. These concepts are described throughout this chapter with an emphasis placed on laser essentials, semiconductors and quantum wells. The chapter concludes with a discussion of generic QC lasers and the process of MBE as an epitaxial growth procedure for these heterostructures. It is necessary to optimize

multiple objectives in the design of a QC laser, so the next chapter discusses multiobjective optimization problems and algorithms used as solutions.

3. Multiobjective Optimization

The simultaneous optimization of multiple objectives is often required when dealing with real-world problems. Conventional optimization techniques, such as simulated annealing and genetic algorithms, were initially designed to solve single-objective problems where the optimality of a solution is shown with one performance measure. These techniques have been extended to and applied to solve multiobjective optimization problems where multiple, often competing, objectives must be optimized to find one, or more solutions.

This chapter introduces multiobjective optimization problems in Section 3.1. It follows with a description of evolutionary algorithms and a more in-depth discussion of genetic algorithms in Section 3.2.1. This is then extended to multiobjective evolutionary algorithms in Section 3.3. The chapter concludes with a discussion of parallel genetic algorithms and an introduction to the most common models.

3.1 Multiobjective Optimization Problems

A multiobjective optimization problem requires the discovery of solutions which strike a balance between the multiple objectives trying to be optimized. Osyczka [62] defined it more formally with an English description:

> *[multiobjective optimization] is the problem of finding a vector of decision variables which satisfy constraints and optimize a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other.*

A formal mathematical model for the general MOP is described in Definition 1 [83]. The set of numbers x_1^*, x_2^*,...,x_n^* is chosen from the set of numbers that satisfy Equations 5 and 6. Every x in the feasible region, Ω, denotes a feasible solution for all objective functions. The function in Equation 7 is a vector representing all the possible objective function values.

Definition 1 - General MOP: Find the vector $\vec{x}^* = [x_1^*, x_2^*,...,x_n^*]$ which satisfies the m inequality constraints:

$$g_i(\vec{x}) \geq 0 i = 1, 2, ..., m \tag{5}$$

the p equality constraints

$$h_i^{\vec{x}} = 0 i = 1, 2, ..., p \tag{6}$$

and optimizes the vector function

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), ..., f_k(\vec{x})]^T \tag{7}$$

In a MOP where solutions are discovered through compromises with several, often competing, objective functions it is important to understand the notion of an optimum solution. The following definitions give a clearer description of the terms commonly used in literature to describe optimality [17].

Definition 2- Pareto Optimality: A point $\vec{x_*} \in \Omega$ is Pareto Optimal if $\forall\ \vec{x} \in \Omega$ and $I = \{1,2,...,k\}$ either,

$$\forall_{i \in I}(f_i(\vec{x}) = f_i(\vec{x^*})) \tag{8}$$

or, there is at least one $i \in I$ such that

$$f_i(\vec{x}) > f_i(\vec{x^*}) \tag{9}$$

Definition 3 - Pareto Dominance: A vector $\vec{u} = (u_1,...,u_k)$ is said to dominate $\vec{v} = (v_1,...,v_k)$ if and only if u is partially less than v, i.e., $\forall\ i \in \{1,...,k\}$, $u_i \leq v_i$ wedge exists $i \in \{1,...,k\}$: u_i ¡ v_i

Definition 4 - Pareto Optimal Set: For a given MOP $\vec{f}(x)$, the Pareto optimal set (P^*) is defined as:

$$P^* := \{x \in \Omega | \neg \exists x' \in \Omega \vec{f}(x') \preceq \vec{f}(x)\} \tag{10}$$

Definition 5 - Pareto Front: For a given MOP $\vec{f}(x)$ and Pareto optimal set P^*, the Pareto front (PF^*) is defined as:

$$PF^* := \{\vec{u} = \vec{f} = (f_1(x), ..., f_k(x)) | x \in P^*\} \tag{11}$$

The Pareto optimal set contains all solutions which are *non-inferior*, *admissible*, or *efficient* and their corresponding *non-dominated* vectors. The Pareto front, PF^* is created when all non-dominated vectors are graphically represented in the objective space.

Many techniques have been used to develop solutions to MOPs. These include such local searching methods as simulated annealing and Tabu search along with stochastic searching techniques like artificial immune systems, genetic algorithms and multiobjective evolutionary algorithms. The latter is chosen for application to the real-world multiobjective QC laser problem. The increase in multiobjective evolutionary algorithms applied to MOPs [80], large objective space, necessity for real-valued representation, relative scarcity of solutions and the rugged landscape made this technique a superior choice.

An overview of evolutionary algorithms, specifics on genetic algorithms and a discussion of multiobjective evolutionary algorithms is included in the next few sections. Additional techniques for solving MOPs, such as simulated annealing, Tabu search and artificial immune systems can be found in Appendix A.

3.2 Evolutionary Algorithms

An evolutionary algorithm's (EA) uniqueness lies in its use of a *population* of solutions for future exploration and exploitation. With a number of solutions the concept of competition can be incorporated. This notion of competition and that of selection simulate those of natural evolution [57]. The pseudocode found in Figure 10 shows the generic steps

followed by an EA. These EAs initialize, evaluate and create new individuals to add to the populations, *P(t)*.

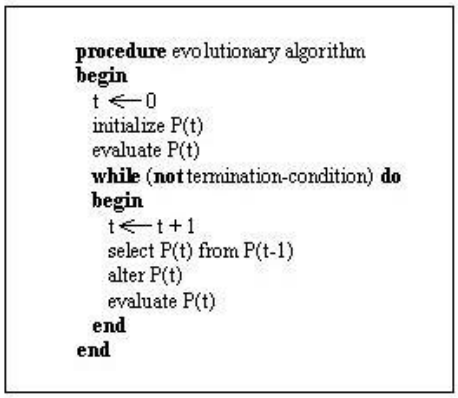

Figure 10 Pseudo Code Representing the Structure of an Evolutionary Algorithm [57]

3.2.1 Genetic Algorithms. The introduction of genetic algorithms occurred in Adaptation in Natural and Artificial Systems written by J. Holland in 1975 [38]. Genetic algorithms, inspired by Darwin, are a way of evolving a solution to the given problem. Because it is often not possible to prove what the optimal solution to a problem is, a genetic algorithm is said to find a suitable, rather than a best, solution. In a genetic algorithm the search space, or set of all feasible solutions, is called the population. Each member of this population is represented as a chromosome, which is made-up of a number of genes, or specific traits. These chromosomes are operated on through crossover and mutation to form a new population of chromosomes, or solutions. The new population is then subjected to a selection function, which chooses the chromosomes that are fit for survival in the next generation.

Crossover uses two "parent" chromosomes and creates two offspring through one or more crossover points. An example of one-point crossover is shown in Figure 11. With only the use of crossover a population risks becoming trapped at a solution which is a local minima. Mutation selects offspring from the crossover operation and randomly changes part of the chromosome.

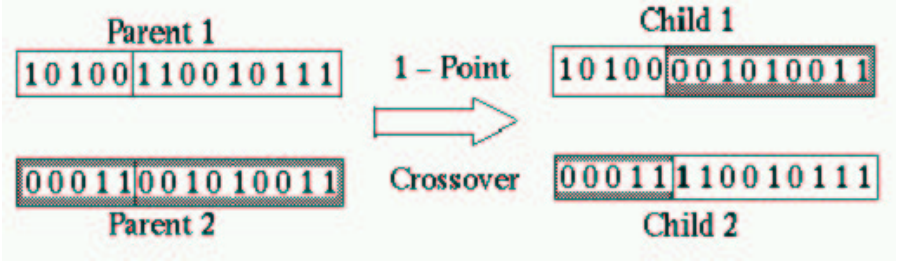

Figure 11 One-point Crossover Operation [39]

There are numerous mechanisms for selecting chromosomes to be included in the crossover operation. The roulette wheel selection method essentially gives each chromosome a probability of being selected. This probability, which relates to the fitness of the chromosome, can be visualized as a portion of a roulette wheel belonging to the chromosome. Rank selection is another method that is also used. Each chromosome in the population is ranked and a fitness value is affixed according to that rank. This ensures that each chromosome has a better chance of being selected than with the roulette wheel method, but it also ensures slower convergence.

Other methods of selection are used to create new population that do not concern choosing parent chromosomes for crossover. Steady-state selection involves directly replacing a certain percentage of chromosomes with low fitness levels with offspring created through crossover. All chromosomes, but those being replaced, survive for the next generation. Elitism is another selection method used that ensures the best chromosomes from the previous population will not be lost when a new generation is formed. This method first copies these "best" chromosomes into the new population before the rest of it is produced [61]. Additional selection methods are available, but the most common have been discussed here.

3.3 Multiobjective Evolutionary Algorithms

A multiobjective evolutionary algorithm (MOEA) is much like an EA, but its defining characteristic is the set of multiple objectives that must be simultaneously optimized. Often MOEAs are a sensible choice of technique for solving MOPs because they combine search with multiobjective decision making [17].

Multiobjective evolutionary algorithms have long been used in the Operations Research community and out of this came a MOEA classification scheme developed by Cohon and Marks (1975) and utilized in [17]. These MOEA algorithm approaches can be decomposed into three categories.

1. *A priori* Preference Articulation.
2. *A posteriori* Preference Articulation.
3. Progressive Preference Articulation.

Following the outline of [17] these three categories are discussed in more detail in Sections 3.3.1, 3.3.2 and 3.3.3 respectively.

3.3.1 A Priori Preference Articulation. An *a priori* technique requires the decision maker (DM) to determine the relative importance of each objective. This effectively turns the MOP into a single-objective problem before the search begins. If the DM has made a poor selection of an objective's importance, then some solutions may be overlooked during the search. Van Veldhuizen breaks the *a priori* techniques into three categories: lexicographic, linear fitness combination and non-linear fitness combination. These techniques are relatively unemployed due to the limits they place on the search space possibly excluding solutions in the true Pareto front [80].

3.3.2 Progressive Preference Articulation. Progressive techniques require that interaction occurs between the DM and the algorithm during the search process which takes place in three steps.

1. Search and find a non-dominated solution.
2. Interact with DM. Modify preferences and objectives if necessary.
3. Repeat 1 and 2 until no further optimization is possible.

This technique is not commonly employed and seen rarely in the literature [80] possibly because so much interaction from the DM is required to find 'good' solutions.

3.3.3 A Posteriori Preference Articulation. With *a posteriori* technique the DM is not required to state objective preference before the search begins. The search is completed and all solutions are offered to the DM for examination. *A posteriori* preference articulation is the favorite classification of MOEA researchers and the most prominent in the literature [80]. This category may be broken down into five separate techniques [47]:

- Independent sampling
- Criterion selection
- Aggregation selection
- Pareto-based selection
- Hybrid selection

Although all *a posteriori* techniques are relatively common the Pareto-based selection method is described here and utilized in this thesis effort. A Pareto-based selection approach utilizes all unique objectives in the problem definition. The Pareto-based fitness of an individual is calculated separately for each objective. These objective fitness values create a solution vector for an individual that determines its inclusion in the Pareto optimal set. Many different algorithms have been developed utilizing the *a posteriori* techniques. A summary of Pareto-based MOEAs, taken from [47] can be found in Appendix B.

3.4 Parallel Genetic Algorithms

Genetic algorithms are ideal for parallelization because of the large area of the domain that must be searched to find a solution. In a serial genetic algorithm these points must be evaluated one at a time, but the independent nature of the evaluation lends itself perfectly to parallelization. In some cases, the reason for parallelizing a genetic algorithm is to reduce the expense of computation and realize some speedup. At other times, parallelization may actually increase the computational time required, possibly due to communication overhead, but, depending on the technique employed, it may reduce the risk of premature convergence. This is often a problem with genetic algorithms even with the addition of mutation as an operator because the mutated chromosome might have such a low fitness

that it isn't selected for inclusion in the next generation [74]. Parallel genetic algorithms can be separated into five categories as shown in Figure 12.

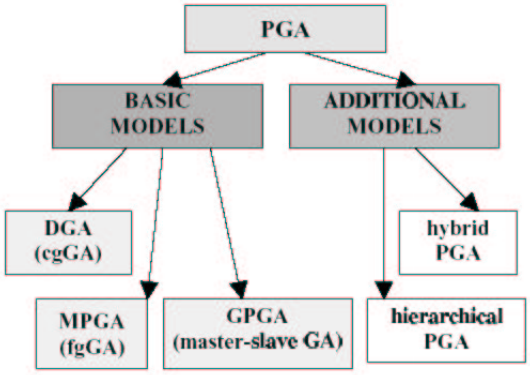

Figure 12 Categories of Parallel Genetic Algorithms

3.4.1 Global Parallel Genetic Algorithm. One of the bottlenecks in a genetic algorithm is the computation time spent evaluating the fitness of each chromosome within a population. Global parallelization attempts to reduce the computation time evaluating individuals and possibly applying appropriate genetic operators in parallel. This method of parallelization preserves the properties of a serial genetic algorithm by maintaining a single population on a "master" processor while utilizing "slave" processors to perform fitness evaluations. Figure 13 is a pictorial representation of this master/slave model of parallelization [33].

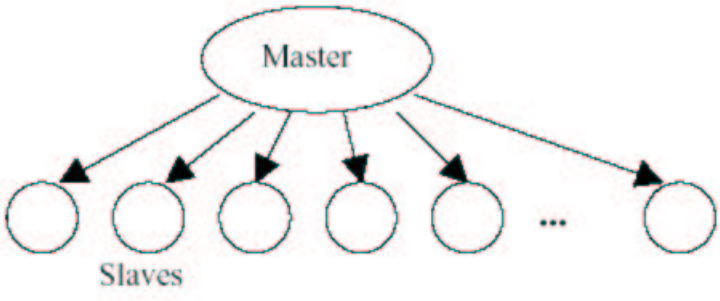

Figure 13 Master/Slave Model of Genetic Algorithm Parallelization [3]

The model acquired its name because selection and mating, crossover and mutation, are still performed globally [11]. In the best case, there would be one processor to evaluate

the fitness of each member of the population. Naturally, this is not always a possibility, so the best speedup can be realized through proper load balancing. Ensuring each processor performs an equal amount of work, not necessarily evaluating an equal number of individuals within the population, achieves the greatest speedup. The speedup for the master/slave parallel GA model can be calculated with Equation 12 where N is the size of the population, p is the number of processors employed, a is the time required for the master processor to perform crossover, mutation and selection while b is the time required for the slave processors to perform the fitness evaluation [74].

$$S(p, N) = \frac{a + bN}{a + \frac{bN}{p}} \tag{12}$$

Because the global parallel genetic algorithm (GPGA), or micro-grain genetic algorithm (mgGA), does not significantly change the structure of the serial GA it does not require any particular network topology for communication. Although, it would benefit slightly, through reduction in communication overhead, if a highly connected network, such as a crossbar switch, was utilized.

Global parallel GAs can proceed in a synchronous or asynchronous manner. Synchronous GPGAs progress exactly the same as a serial genetic algorithm by waiting for the fitness values of every individual in a population to be evaluated before proceeding to the next generation [34]. In an asynchronous GPGA the master does not wait for all slave processors to return the fitness value evaluations. The progress of the algorithm is handled in many different manners [11].

3.4.2 Coarse-Grained Parallel Genetic Algorithms. The coarse-grained genetic algorithm (cgGA) is often referred to, in literature, as the island model [35], or a distributed genetic algorithm (DGA) [33]. Wherein a GPGA the entire population is maintained on the master processor, in a cgGA the population is split into many subpopulations, or demes, that are operated upon in parallel. Migration occurs between the subpopulations through a specific selection policy and at some predetermined migration rate. Three different timing constraints can be used to control the method of migration. In the case of

isolated island GAs there is no migration occurring. For a synchronous cgGA each subpopulation, existing on a separate processor, evolves at its own rate, but mutation occurs at a constant rate which is most likely at the speed of the slowest processor. Asynchronous cgGAs allow migration to be triggered not at the speed of a processor, but by the occurrence of a particular event [53]. A cgGA is difficult to implement effectively because these previously mentioned constraints require communication to occur between processors within the model. This communication pattern is defined by the network topology of the processors being utilized. Often, implementations of cgGAs adopt a static topology such as the hypercube, seen in Figure 14 and used in [76] and [19].

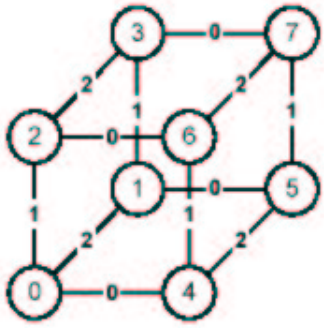

Figure 14 Static Hypercube Topology [51]

Other cgGA implementations, [35], utilize a ring topology, shown in Figure 15. In some cases it may be beneficial to use a dynamic network topology while evolving a cgGA to avoid an incompatible individual migrating into a subpopulation and either being ignored or dominating this subpopulation [53].

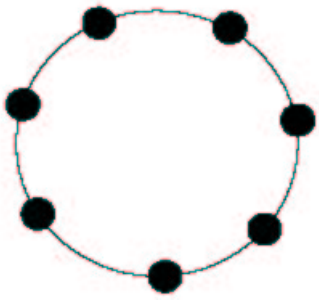

Figure 15 Static Ring Topology [52]

3.4.3 Fine-Grained Parallel Genetic Algorithms. In a fine-grained genetic algorithm (fgGA), often called a massively parallel genetic algorithm (MPGA), individuals in a population are usually assigned one to a processor. Each of these individuals are part of one large population rather than the subpopulations created in a cgGA. The difference is that, despite the global population, the individuals only interact locally. Each individual is part of a neighborhood, or deme, where the demes are overlapping [2]. Neighborhoods can be defined in many different ways, three of which are denoted in Figure 16. The selection and mating operators are restricted to the neighborhood, but the policy of overlapping neighborhoods allows individuals the freedom to move through the entire population [33]. The main purpose of this design is to allow migration of genetic information throughout the population at a pace slower than that found in GPGA, but without the need for a specified migration policy as in a cgGA. This rate of migration is determined by the neighborhood structure which is often defined by the network topology. A network with a high diameter will slow migration considerably and generally solve the problem of premature convergence, but may require too many generations to find a solution [74].

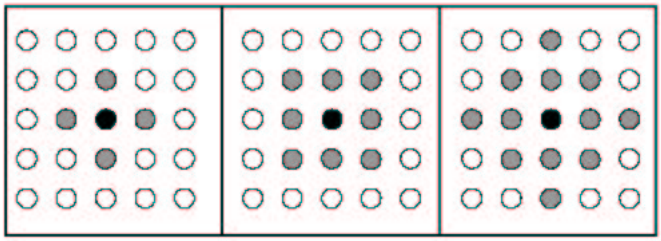

Figure 16 Neighborhood Definitions for fgGAs

Table 3 displays specific implementations of parallel genetic algorithms, the type of parallelization utilized and the network topology employed.

Where parallel GAs are concerned it is not always possible to create a situation where a one-to-one relationship exists between the number of processors available and the number of individuals that need to be evaluated. In these instances proper load balancing is the key. For a detailed description of load balancing techniques see Appendix C.

Table 3 **Specific Implementations of Parallel Genetic Algorithms**

Parallel GA	Kind of Parallelism	Network Topology
ASPARAGOS	Fine-grain. Uses Hill-Climbing.	Ladder
CoPDEB	Coarse-grain. Applies diff. operators to sub-pops.	Fully Connected
DGENESIS 1.0	Coarse-grain. Migrations among sub-pops.	Any Desired
ECO-GA	Fine-grain.	Grid
EnGENEer	Global parallelization.	Master/Slave
GALOPPS 3.1	Coarse-grain.	Any Desired
GAMAS	Coarse-grain. Uses four species of strings.	Fixed Hierarchy
GDGA	Coarse-grain. Explicit exploration/exploitation.	Hypercube
GENITOR II	Coarse-grain. Unique crossover operator.	Ring
HSDGA	Hierarchical coarse and fine-grain GA.	Ring, Tree and Star
PARAGENESIS	Coarse-grain.	Multiple
PeGAsuS	Coarse or fine-grain. MIMD	Multiple
PGA 2.5	Coarse-grain. Migrations among sub-pops.	Multiple
PGAPack	Global parallelization.	Master/Slave
RPL2	Coarse-grain.	Any Desired
SGA-Cube	Coarse-grain. Made for the nCUBE2	Hypercube

3.5 Summary

Mulitobjective optimization problems require solutions to be compromises between more than one objective. In this chapter numerous approaches for solution discovery in MOPs, such as EAs, GAs and MOEAs, were discussed. A multiobjective evolutionary algorithm titled General Multiobjective Parallel GA (GenMOP) is chosen for application to the QC laser optimization problem. Further details of the problem domain are discussed in the following chapter while a comprehensive description of GEenMOP is given in Chapter 5.

4. Design of a Quantum Cascade Laser Model

The process of designing a terahertz emitter utilizing the principles of quantum cascade lasers began following the work of Menon [55]. With sponsorship from the Air Force Research Laboratory Sensors Directorate Dr. Ram-Mohan, Jacob Gagnon and Alexi Girgis designed a software model that is utilized in place of device fabrication to simulate a quantum cascade laser.

This chapter discusses the design principles of quantum cascade lasers in Section 4.1 with a description of the software model shown in Figure 19. The sections following that diagram introduce the software modules by discussing the underlying mathematics required for computing the feasibility of a combination of laser parameters. The chapter is concluded with a discussion of the difficulties in creating a model within software.

4.1 Quantum Cascade Laser Design

A defining property of the quantum cascade (QC) laser design is the cascading effect formed through a bias applied to a multiple quantum well structure. A combination of ten to one hundred of these structures containing both active and collector/injection regions forms the semiconductor device. Figure 17 is a three energy level quantum well structure diagramming the cascading scheme within a QC laser and representing the structure modelled for the simulation. The transfer of electrons from one quantum well active region structure through the collector/injection region to the next active region structure is highlighted.

Photon emission occurs through an intersubband transition, shown in Figure 18, between quantum confined levels within the same conduction band in these structures. Intersubband transitions can be either radiative or non-radiative. A radiative transition occurs whenever an electron transitions from a higher to a lower energy level and emits a photon. This is illustrated in Figure 17 between energy levels three, E_3, and two, E_2, where hv denotes the energy of the photon equal to $E_3 - E_2$. Non-radiative transitions occur when the electron experiences electron-electron scattering, electron-confined phonon

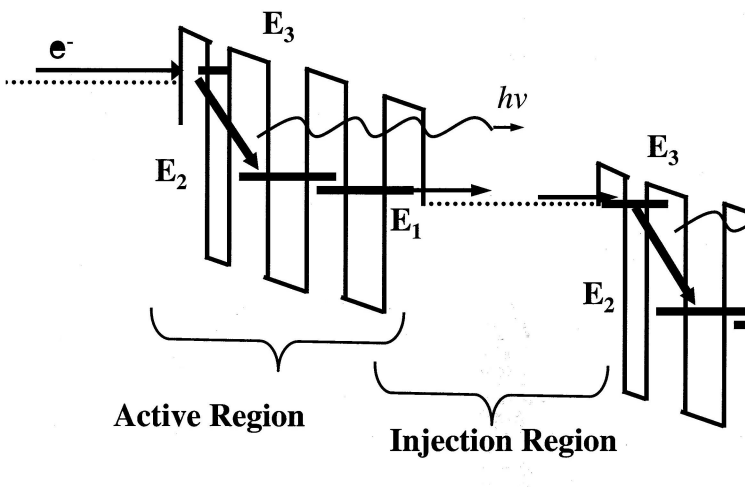

Figure 17 Schematic of the operation of a quantum cascade laser. [55]

scattering, or electron-interface scattering. These are explained in more detail in Sections 4.5.1, 4.5.2 and 4.5.3 respectively.

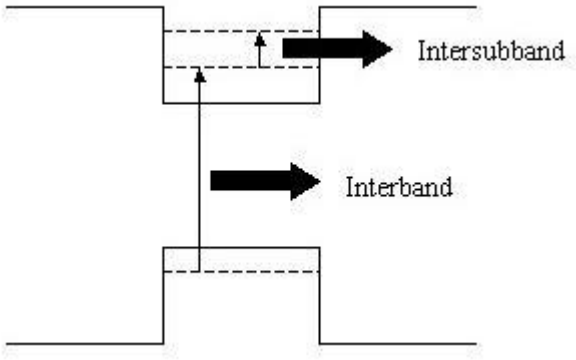

Figure 18 Demonstrating the Distinction Between Interband and Intersubband Transitions. [55]

To guarantee lasing occurs within the QC laser structure the radiative recombination mechanism must dominate all of the non-radiative recombination mechanisms. A population inversion is created when the rate of radiative recombination between energy level three and two is greater than the rate of non-radiative electron transitions between energy levels two and one. Equation 13 describes the population inversion under steady-state conditions. Here $\frac{1}{\tau_3} = \frac{1}{\tau_{31}} + \frac{1}{\tau_{32}}$, J represents the current density, $\bar{g}$ the intersubband gain

in the active region, e denotes the charge of an electron and S the photon density per unit area [55].

$$n_3 - n_2 = \frac{\tau_3}{\tau_{32}} (\tau_{32} - \tau_{21}) \left[\frac{J}{e} - \bar{g} S \left(1 + \frac{\tau_{21}}{\left(\frac{\tau_3}{\tau_{32}} \right) (\tau_{32} - \tau_{21})} \right) \right] \tag{13}$$

The transition times τ_{ij} in Equation 13 are computed as

$$\frac{1}{\tau_{32}} = \frac{1}{(\tau_{32})_{rad}} + \frac{1}{(\tau_{32})_{ac}} + \frac{1}{(\tau_{32})_{e-e}} \tag{14}$$

where rad is the μs range, ac is the acoustic phonon mediated lifetime and $e - e$ is the electron scattering.

Through the use of interface phonons, phonon mediation, whose lifetime is represented in Equation 14 by τ_{32ac}, was chosen as the population inversion mechanism for modelling purposes because continuous wave operation is desired at or near room temperature and this depopulation mechanism has been shown to increase in strength at higher temperatures [55]. A bias is applied to the semiconductor structure to ensure the ground state, or lowest energy level, of an injection region equates to the highest energy level in the next active region. This phenomenon is shown in Figure 17.

The quantum cascade laser device is modelled through bandgap engineering and gain optimization techniques. The three-layer design of the active region of this semiconductor device, described in more detail in Section 4.2, is constructed from both GaAs and AlGaAs. A comprehensive software model was designed at Worchester Polytechnic Institute. Figure 19 depicts the calculations required to evaluate and identify laser parameters creating solutions. The remaining sections in this chapter mathematically describe the quantum cascade laser model. The input to the laser model consists of parameters defined by the GA for each individual and the additional parameters outlined Appendix D.

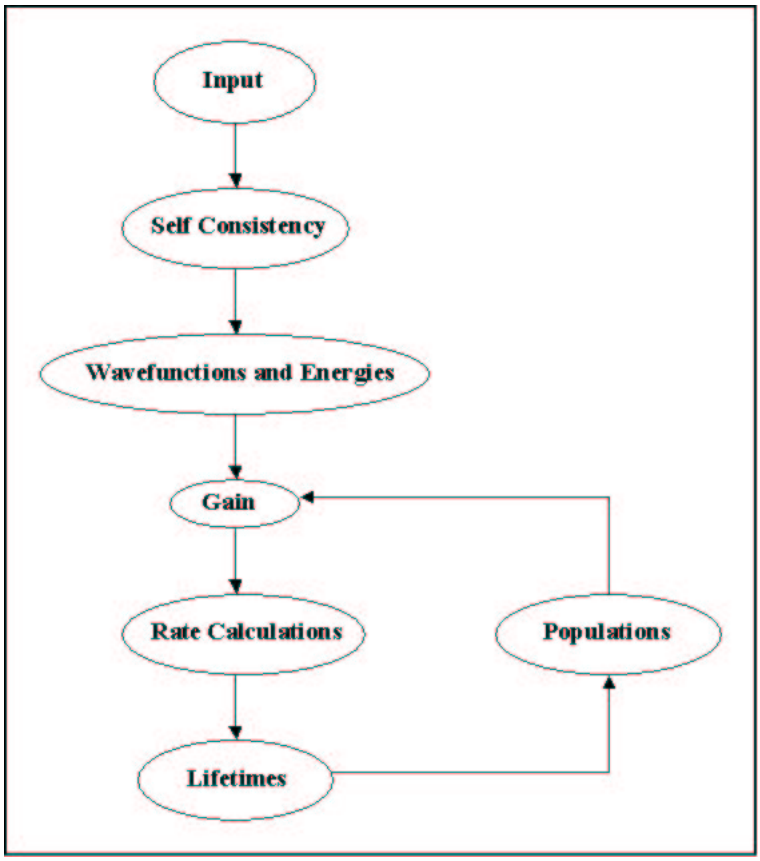

Figure 19 Flow Diagram of the Quantum Cascade Laser Model

4.2 GaAs/AlGaAs Quantum Cascade Laser

A one-dimensional quantum potential well is formed with one layer of Gallium Arsenide (GaAs) surrounded by two barrier layers of $Al_xGa_{1-x}As$, where x represents the impurity concentration at a value of .3 for this model. This quantum well heterostructure is used for the confinement of carrier electrons. A modification in the layer thickness of the quantum well changes the relative energies of the quantized electron energy levels while an adjustment in the alloy concentrations within the materials changes the barrier heights. This in-turn changes the energy of the quantized electron levels.

Through the use of one layer of GaAs and two barrier layers of $Al_xGa_{1-x}As$ a general quantum heterostructure that forms a one-dimensional quantum well for confinement of electronic carriers is designed. The layer thicknesses determine the depth of the quantum wells while modification of the alloy concentrations within the materials changes the barrier

heights, so a desired quantum structure can be modelled. The importance of intersubband transitions in forming the perfect model was discussed above.

GaAs, a compound semiconductor, crystallizes in a structure almost identical to the diamond lattice described in Chapter 2. The difference lies in this crystal consisting of gallium atoms that are each bonded to four arsenic atoms. The zinc blende lattice, shown in Figure 20 separates its lattice sites equally between the two atoms.

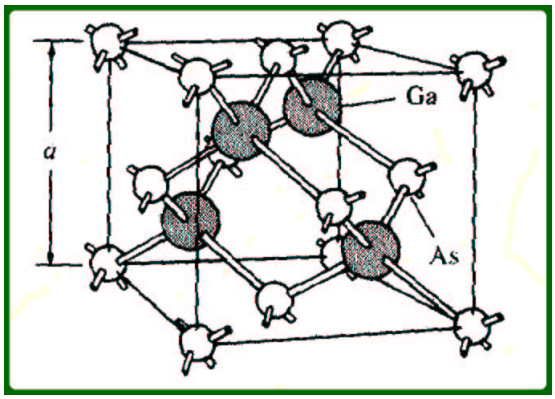

Figure 20 Zinc Blende Lattice Unit Cell [65]

This crystalline composition in a lattice structure determines the energy levels at which lasing occurs. The closely spaced energy levels within the material form an energy band. These bands are separated from each other with energy bandgaps. The electrons in the highest almost-filled energy band and those in the lowest partially-filled band determine the properties of the semiconductor. This bandgap within GaAs is labelled direct meaning it allows for more efficient absorption and emission of light [84].

The material GaAs exhibits superior electron transport properties and special optical properties as compared to other compound semiconductors. Terahertz emitters created with GaAs are generally fabricated using solid source molecular beam epitaxy on n^+ GaAs substrates [56]. A more detailed description of this process can be found in Section 2.6.

4.3 Wavefunction and Energy Level Calculations

The structure must be specifically designed, so that the energy spacing between energy levels two and three corresponds to the energy of radiation in the terahertz frequency

range. This spacing ensures that during transition an electron emits a photon with energy $E_3 - E_2$ equal to the energy of a terahertz photon. Additionally, the energy level spacing between levels two and one must equal the energy of an interface phonon. To allow periodic transitions the first energy level is designed, so that under a voltage bias it equals the third energy level for the next active region period. This allows the electronic carriers to be collected and subsequently injected into the next period [29].

The time-independent Schrödinger's equation for a quantum heterostructure, shown in Equation 15, must be solved to determine the energy levels described above as well as the wavefunctions which are utilized during the lifetime calculations. Here $V(z)$ represents the potential energy, r is the wavefunction and E is the energy. Because the Schrödinger equation cannot be solved analytically another numerical technique such as the transfer matrix method, or the finite element method must be exploited [56].

$$-\frac{\hbar^2}{2m_i^*}\nabla^2\psi(r) + V(z)\psi_i(r) = E\psi_i(r) \tag{15}$$

4.3.1 Finite Element Method. Finite element analysis is chosen over the transfer matrix method. The finite element analysis method is used to find approximate solutions to problems in which no closed-form solution exists. An infinite number of degrees of freedom are present in these continuous domain problems, so the solution must be found by reducing this to a finite number of unknowns [43]. The degree of the approximation relies heavily on the first step of the finite element method (FEM), domain discretization. The chosen number of elements as a problem representation not only depends on the numerical solution accuracy necessary, but may also be influenced by storage limitations, or excessive computation times. After discretization interpolation functions must be selected to approximate the solution within an element. The most commonly used function is a first-order linear equation because the accuracy obtained from higher-order polynomials often does not justify their complexity [44]. A system of equations is designed and solved to find the finite number of unknown values in the problem.

4.4 Gain Calculation

For some laser productions the goal is to maximize the gain as proof of superior performance. As shown in [55] the gain in our three energy-level QC laser structure is represented by Equation 16.

$$g(\lambda) = \frac{\Delta N}{L_p} \frac{\lambda^2}{4\pi n_2 \tau_{spon}} \frac{T_3}{1 + 4\pi^2 c^2 (\lambda^{-1} - (\frac{2\pi\hbar c}{\Delta E_{32}})^{-1})^2 T_2^3} \tag{16}$$

where ΔN represents the population inversion carrier density while L_p denotes the length of one period in the structure. T_3 is the intersubband scattering time while n indicates the average refractive index. A more complete description of Equation 16 and its parameters can be found in [55].

4.5 Rate Equations

The non-radiative transitions mentioned above are associated with interface phonon mediation. These are described in detail in Sections 4.5.1, 4.5.2 and 4.5.3. More details on the derivations of the scattering effects and the functional forms of the lifetimes can be found in [29].

4.5.1 Electron-Electron Scattering Rate. An electron-electron scattering lifetime is determined for each electron by summing over the initial in-plane momentum states and then accounting for the Fermi blocking effects. The final derivation can be seen in Equation 17.

$$\frac{1}{\tau} = 2 * \frac{\mathcal{A} m_i^*}{(2\pi)^2 h^2} \int_0^{\inf} dE_{i,||} \int_0^{2\pi} d\theta \frac{F_i(\vec{k_i}) 1}{\tau(\vec{k_i}) \mathcal{A}_{||} n_i} \tag{17}$$

where E is the center of mass energy for each electron, $\tau(\vec{k_i})$ represents the scattering lifetime as a function of $\vec{k_i}$, m is the effective mass, A denotes the vector potential, $F(\vec{k_i})$ is a form factor and $\frac{1}{A_{||} n_i}$ is used to average over the initial states.

Computing electron-electron scattering utilizing Equation 17 is computationally expensive. To increase evaluation during integration an approximation of the form factor

is precalculated. Further explanation of this approximation and derivations leading to Equation 17 are available in [29].

4.5.2 Electron-Confined Phonon Scattering Rate. The total electron-confined phonon scattering rate versus the width of a quantum well is shown in Equation 18.

$$W_{ij} = \frac{e^2 \mathcal{A}_{||} m_j^* 2\pi m_i^*}{2\pi h (2\pi)^2 h^2 h^2} \left(\frac{n+1}{n}\right) \int_0^{E_{fi}+5k_bT-E_{i,0}} dE_{i,||} \int_0^{2\pi} d\theta * \sum_{\alpha i} \mid N_{\alpha i}\left(q_{||}\left(\theta,(E_{i,||})\right)\right) G_{ij\alpha_i} \mid^2 F(\vec{k}_i)\left(1-F(\vec{k}_j)\right) \frac{2}{\mathcal{A}_{||} n_i} \tag{18}$$

where G represents the overlap of the initial and final wavefunctions with the phonon potential, N is the mode normalization, $q_{||}$ denotes the magnitude of the momentum transfer and $\frac{1}{A_{||}n_i}$ is the averaging over the initial states.

4.5.3 Electron-Interface Scattering Rate. To calculate the electron-interface scattering rate the functional form of the interface modes and the normal mode frequencies must first be found. Following this Fermi's Golden Rule is applied [36] in addition to Fermi blocking effects. The variables of integration are changed and the energy conserving delta function is expanded. The electron-interface scattering is shown in Equation 19.

$$\frac{e^2 m_j^*}{2\pi h^2}\ sum_{\alpha i} \int d\theta dE_{j,||} \left(\frac{n(\omega_{\alpha i}(q_{||}))+1}{n(\omega_{alphai}(q_{||}))}\right) \mid N_{\alpha i}(q_{||}) G_{ij\alpha i}(q||) \mid^2 \delta(E_{j,0} + \frac{h^2 k_j^2}{2m_j^*} - E_{i,0} - \frac{h^2 k_i^2}{2m_i^*} \pm h\omega_{\alpha i}(q_{||}(k_i,k_j)))(1-F(\vec{k}_j)) \tag{19}$$

where G represents the overlap integral, N is the mode normalization, $q_{||}$ is the magnitude of the momentum transfer and ω is the effective angular frequency. Details leading up to the derivations of Equations 17, 18 and 19 can be found in [29].

4.6 Software Details

The QC laser model software is designed in C++. It utilizes five libraries for dense and sparse matrix manipulations. Standing alone it has the ability to analyze one set of laser parameters for the feasibility of a solution. As input it requires a laser parameter file, an example can be found in Appendix D. This input file contains information about the number of layers within a structure, the material, the impurity concentration, their widths, donor density and donor energy. Additionally, it provides the bias in kV/cm, the current density in kA/cm^2, the temperature in Kelvin and the boundary conditions. An option for self consistency is given with tolerances. Gain calculations may be turned on, or off. If they are on, then the number of energy levels must be specified along with the populations for these levels. Scattering rate options are also available. With electron-electron scattering rates the method must be defined as either Gauss, or Monte Carlo. The number of events are specified and the the tolerances are given. For electron confined phonon scattering and electron interface phonon scattering the options are either turned on, or off and no additional parameters need be specified.

Beyond these calculations the model also offers output options to create visualizations after execution. These options include plotting the energies and wavefunctions, the self consistency, dispersion, omegas and gain. See Appendix D for a detailed laser parameter input file.

At the beginning of the integration between the QC laser model software and GenMOP some data structure problems were experienced. These surfaced through the use of a new compiler necessary for GenMOP's parallelization. The issues arose during matrix manipulations involving the matrix libraries. Additionally, the number of exact calculations initially required to detect a solution's feasibility required four to seven hours. To remove this bottleneck the self consistency was eliminated and additional mathematical approximations made. Making these changes resolved the difficulties and integration with GenMOP was successful.

4.7 Summary

The quantum cascade laser structure was discussed in detail with an introduction to intersubband transitions and interface phonon mediation used for population inversion. The software model is outlined and shown in Figure 19. The specific material chosen for the model and the mathematics necessary for evaluation are given. This leads us to the following chapter where details of the algorithm employed are offered.

5. Algorithm Design

The ultimate goal of this research is to integrate a MOEA to the QC laser model software that creates both an effective and efficient process for finding solutions. For optimization of the QC laser operating in the terahertz frequency range a multi-objective algorithm which can handle real values is required. Parallelization is desired because of the intensive computation required to correctly simulate the operation of the QC laser given specific parameters. Due to these constraints a multiobjective genetic algorithm modified to run in parallel was chosen. The particular algorithm chosen was the General Multiobjective Parallel Genetic Algorithm (GenMOP) designed at the Air Force Institute of Technology [30] for solution to a groundwater remediation problem.

The algorithm is introduced in Section 5.1. A discussion of its equivalence class sharing technique for filling the mating pool is introduced in Section 5.2. The crossover and mutation operators employed are described in Sections 5.3 and 5.4, respectively. The chapter concludes with a discussion of the software model developed.

5.1 General Multiobjective Parallel Genetic Algorithm

GenMOP is a pareto-based algorithm that utilizes real values for crossover and mutation operators. Additionally, the algorithm employs fitness sharing through a niche radius. The original genetic algorithm was designed to find solutions to an aggregated-objective groundwater remediation problem [30], but has been modified to perform optimization for a QC laser.

The individual chromosomes are encoded with real-values denoting the temperature, bias, current density, layer thickness and donor density of a particular laser. Auxiliary genes are associated with the individual chromosomes to define fitness values and Pareto ranking. All GA operations, to include mutation and crossover, are performed solely on the chromosome without interaction from the auxiliary genes.

There are five parameters that the user has the ability to specify with GenMOP: mutation probability, p_m, initial population size, Pop_0, number of generations, N, mating

Table 4 **Default Values for GA Parameters**

GenMOP Parameter	Default Setting
Mutation Probability	.02
Initial Population Size	20
Number of Generations	10
Mating Pool Size	10
Niche Radius	.02

pool size, *MP*, and the niche radius, σ_{share}. Default settings for these parameters can be seen in Table 4.

If no input file is specified to begin GenMOP execution a population of size Pop_0 is randomly initialized. Instead of utilizing a repair function after new individuals are created, all parameters have minimum and maximum values that constrain the chromosome construction. These initial chromosomes are stored in the cumulative population, Pop_{cum}. Each individual within this population is evaluated for its fitness and then these fitnesses are granted a Pareto rank. This Pareto rank corresponds to the number of chromosomes that dominate the particular individual. A non-dominated chromosome would hold the Pareto rank of zero. This Pareto ranking scheme was developed by Fonseca and Fleming [26].

Once Pareto ranking has terminated, selection for the mating pool begins. Individuals are selected first based on their Pareto rank. When more individuals are present in a particular rank than spaces left in the mating pool, defined by *MP*, then the equivalence class sharing technique [42] is used to measure crowding within the objective space. Chromosomes relating to less crowded areas of the objective space will be chosen for the mating pool to help preserve diversity within the population.

5.2 Equivalence Class Sharing

The equivalence class sharing technique is used when the number of k-rank chromosomes exceeds the space remaining in the mating pool. In this case we let x_i denote a specific k-rank chromosome where $i = \{1,2,3,\ldots,\#\ of\ k\text{-}rank\ chromosomes\}$. Additionally, x_j = *any chromosome in* Pop_{cum}, where $j = \{1,2,3,\ldots,\#\ of\ chromosomes\ in\ Pop_{cum}\}$.

The maximum and minimum values for objective functions f_1 and f_2 are found in Pop_{cum}. Utilizing the maximum and minimum fitness values, normalization $\forall\ x_i \in$ *population of k-rank chromosomes* and $\forall\ x_j \in Pop_{cum}$ occurs through the following equations derived for [48]:

$$f_{1_i}' = (f_{1i} - f_{1min}) \,/\, (f_{1max} - f_{1min})$$
$$f_{2_i}' = (f_{2i} - f_{2min}) \,/\, (f_{2max} - f_{2min})$$
$$f_{1_j}' = (f_{1j} - f_{1min}) \,/\, (f_{1max} - f_{1min})$$
$$f_{2_j}' = (f_{2j} - f_{2min}) \,/\, (f_{2max} - f_{2min})$$

where

$f_{1_i}' =$ *dimensionless value of* f_1 *based on* $x_i \in$ {*population of k-rank chromosomes*}

$f_{2_i}' =$ *dimensionless value of* f_2 *based on* $x_i \in$ {*population of k-rank chromosomes*}

$f_{1_j}' =$ *dimensionless value of* f_1 *based on* $x_i \in Pop_{cum}$

$f_{2_j}' =$ *dimensionless value of* f_2 *based on* $x_i \in Pop_{cum}$

$f_{1i} =$ *value of* f_1 *based on* $x_i \in$ {*population of k-rank chromosomes*}

$f_{2i} =$ *value of* f_2 *based on* $x_i \in$ {*population of k-rank chromosomes*}

$f_{1j} =$ *value of* f_1 *based on* $x_j \in Pop_{cum}$

$f_{2j} =$ *value of* f_2 *based on* $x_j \in Pop_{cum}$

$f_{1min} =$ *minimum value of* $f_1 \in Pop_{cum}$

$f_{1max} =$ *maximum value of* $f_1 \in Pop_{cum}$

$f_{2min} =$ *minimum value of* $f_2 \in Pop_{cum}$

$f_{2max} =$ *maximum value of* $f_2 \in Pop_{cum}$

Following the normalization of the fitness values for both objective functions the distance between points (f_{1_i}', f_{2_i}') and (f_{1_j}', f_{2_j}') is calculated using Equation 20.

$$d_{ij} = [(f_1`_i - f_1`_j)^2 + (f_2`_i - f_2`_j)^2]^{1/2} \tag{20}$$

The previously calculated distance, d_{ij}, and the user-defined niche radius, σ_{share} are used to compute the sharing function shown in Equation 21. The sharing function, by measuring the proximity of point $(f_1{}'_i,\ f_2{}'_i)$ to $(f_1{}'_j,\ f_2{}'_j)$ and taking into account σ_{share}, determines the amount of crowding within the defined circle. The niche radius identifies the radius of the circle around a specified point. Any point found within the circle contributes to crowding with the points overlapping when $Sh(d_{ij}) = 1$. If point $(f_1{}'_j,\ f_2{}'_j)$ rests outside the circle surrounding $(f_1{}'_i,\ f_2{}'_i)$, then $Sh(d_{ij}) = 0$ and there is no crowding.

$$Sh(d_{ij}) = \begin{cases} 1 - d_{ij}/\sigma_{share} & \text{for } d_{ij} \leq \sigma_{share}, \\ 0 & \text{for } d_{ij} > \sigma_{share} \end{cases} \tag{21}$$

In order to fill the remaining space in the mating pool the niche count must be determined for each k-rank chromosome. The niche count, m_i, for chromosome $x_i \in \{$*population of k-rank chromosomes*$\}$ is calculated with Equation 22.

$$m_i = \Sigma_{x_j \in Pop_{cum}} Sh(d_{ij}) \tag{22}$$

The niche count is simply a summation of all sharing function values attributed to each chromosome within Pop_{cum} with respect to a particular $x_i \in \{$*population of k-rank chromosomes*$\}$. A higher m_i denotes greater crowding for an individual. The remaining space in the mating pool is filled with chromosomes having the lowest m_i values available. These individuals are chosen to preserve diversity within the population which leads to solutions occupying less crowded areas of the Pareto front.

5.3 Crossover

The entire mating pool is now subjected to crossover and mutation operators developed in [57]. Crossover occurs in one of four ways $\forall\ x_i \in$ the mating pool, where $i =$

$\{1,2,3,...,|MP|\}$. For the first three types of crossover described below a second individual, x_r, is chosen at random from MP to be crossed with x_i.

1) *Whole Arithmetical Crossover*: All genes of x_i and x_r are linearly combined to form chromosomes x_1 and x_2. GenMOP retains x_1 and discards x_2.

2) *Simple Crossover*: One gene is chosen in both x_i and x_r and swapped to form chromosomes x_1 and x_2. GenMOP retains x_1 and discards x_2.

3) *Heuristic Crossover*: Individuals x_i and x_r are combined to form one individual $\ni x_1 = R \cdot (x_r - x_i) + x_r$, where R is a uniform random number between zero and one and the rank of $x_r \leq x_i$.

4) *Pool Crossover*: Randomly chooses genes from individuals in the mating pool and combines them to create x_1.

The type of crossover to be performed on the two individuals is chosen based upon an adaptive probability distribution. Each of the four crossover types described above begins with the same probability of being chosen. As the algorithm progresses through generations these probabilities are adapted through the fitness of the individuals they create. If the newly created individual dominates x_i, then the fitness of the newer individual was increased over the previous through use of this particular crossover operator. Consequently, because of the success of the new individual the crossover operator's selection probability increases.

5.4 Mutation

The new individuals created through a crossover operation are now subject to mutation with a probability defined by the user, p_m. If a number, n is randomly selected from a uniform distribution, so that $0 < n < 1$ and $n < p_m$, then one of three mutation operators described below is chosen to perform on the individual.

1) *Uniform Mutation*: Chooses a gene existing in the chromosome to reset to a random value within its specified ranges.

2) *Boundary Mutation*: Chooses a gene existing in the chromosome to reset to either its maximum or minimum value.

3) *Non-uniform Mutation*: Chooses a gene to modify by some random value decreases probabilistically, until it equals zero, as the generation number approaches the maximum generations.

The mutation operator is selected using the same adaptive probability distribution described previously for crossover operations. Between these two operators a new population is developed, Pop_{new} which is equal to $|MP|$. Each individual in Pop_{new} is evaluated for fitness and then placed in Pop_{cum}, where chromosomes from all generations reside.

5.5 Software Details

The GenMOP software follows, as Figure 21 illustrates, from the required input to population initialization, through a preliminary evaluation, ranking and normalization of this population. If the maximum number of user specified generations has not been reached, then GenMOP fills the mating pool with individuals from the cumulative population maintaining the highest rank. Crossover and mutation are performed on these individuals followed by an evaluation. Once all the individuals are returned to the cumulative population ranking takes place. The children are saved in an output file and if the maximum number of generations is reached the program terminates and writes all the individuals in the cumulative population to an additional output file. If the maximum number of generations has not been reached, then the GA loops back, refills the mating pool, performs crossover and mutation, evaluates the new individuals, places them back in the cumulative population and ranks the whole population. This loop continues until the user-defined maximum number of generations is reached and the program is terminated.

It is necessary for the user to be able to define both the number of layers in a structure being simulated and the target frequency. These two pieces of information are input on the command line along with names for the output files created and input files for both the laser and GA parameters. The input file for the laser parameters was discussed in Section 4.6 while the GA parameter input file must include all the items found in Table 4.

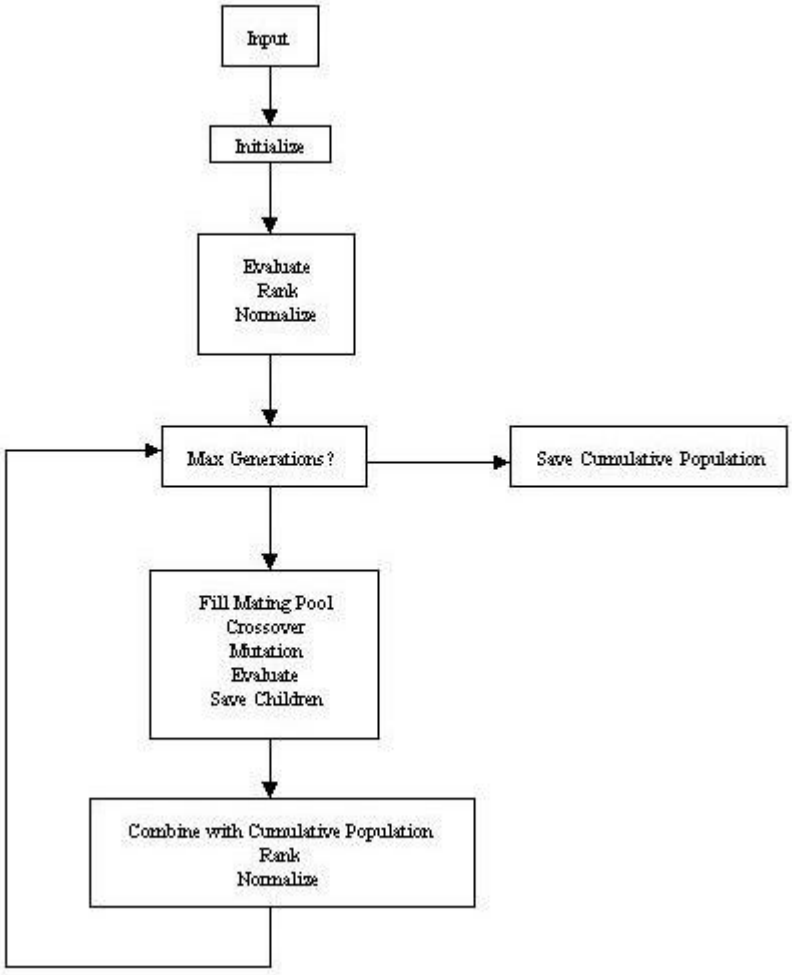

Figure 21 Flow Diagram of GenMOP code.

Before integration changes were made to the GenMOP software to give the user more flexibility by allowing for more variables, those discussed above, to be entered on the command line. This was done for ease of use with scalability in mind. Additionally, the chromosome structure and auxilary genes were designed to be dynamically defined according to the number of layers in the structure being simulated. To avoid having to create a repair function for the algorithm maximums and minimums were identified for each gene within a chromosome and checked before any laser calculations were performed. Because the QC laser problem is defined as a maximization MOP the Pareto ranking scheme of GenMOP needed to be altered. Objective functions were defined to evaluate the

quality of each individual created by the GA and the algorithm was integrated with the QC laser model software.

5.6 Summary

The General Multiobjective Parallel GA is introduced in this chapter. The crossover and mutation operators utilized are explained. The equivalence class sharing technique is defined as it is used for selection of individuals. These methods are created in a C^{++} program which is integrated with the QC laser model described in Chapter 4. The experiments designed for testing the efficiency and effectiveness of the algorithm for this MOP are outlined in the following chapter.

6. Design of Experiments

At the culmination of this research it is desired that a correct model of a quantum cascade laser operating in the terahertz frequency range, described in Chapter 4, be integrated with a multiobjective genetic algorithm, described in Chapter 5, to generate multiple possible parameter combinations that create unique solutions. These solutions are found through a computerized QC laser simulation and MOEA optimization designed specifically for this purpose.

This chapter describes the parallelization of GenMOP in Section 6.1 along with a discussion of the importance of efficiency and effectiveness for algorithm design. The discussion of efficiency and effectiveness is expanded and metrics for evaluation are offered in Sections 6.2 and 6.3, respectively. Within each of these sections an overview of experiments is given with details on the setup and execution of experiments. The chapter concludes with a summary of the discussion.

6.1 Parallel GenMOP

One of the greatest bottlenecks in a genetic algorithm is the computation time spent evaluating the fitness of each chromosome within a population. Additionally, the complexity of the QC laser model suggests that the fitness evaluation time for this simulation is going to be lengthy, so global parallelization is utilized. Global parallelization attempts to reduce the computation time evaluating individuals and possibly applying appropriate genetic operators in parallel. This method of parallelization preserves the properties of a serial genetic algorithm by maintaining a single population on a "master" processor while utilizing "slave" processors to perform fitness evaluations [33]. This parallel genetic algorithm model is described in detail in Section 7.2.3.

The model, also known as a global parallel genetic algorithm (GPGA) model, acquired its name because selection, mating, crossover and mutation are still performed globally [11]. In a best-case scenario, and for the experiments performed here, the fitness for each member of the population is calculated on a separate processor. Naturally, it is not always possible to create this situation, so for those instances where a one-to-one rela-

tionship cannot be obtained proper load balancing derives the most speedup. This occurs by ensuring each processor performs an equal amount of work, not necessarily evaluating an equal number of individuals within the population.

Because the global parallel genetic algorithm, or micro-grain genetic algorithm (mgGA) as it is sometimes referred to, does not significantly change the structure of the serial GA it does not require any particular network topology for communication. Although, it would benefit slightly through reduction in communication overhead if a highly connected network, such as a crossbar switch, was utilized.

GPGAs can proceed in either a synchronous or asynchronous manner. In an asynchronous GPGA the master does not wait for all slave processors to return the fitness value evaluations before proceeding. The progress of this type of algorithm can be handled in many different manners described in further detail in [11] and is much more complex than a synchronous GPGA. Because of the complexity involved in designing an asynchronous GPGA and the inability to predict its performance improvement GenMOP follows a synchronous GPGA routine. Synchronous GPGAs operate exactly the same as a serial genetic algorithm by waiting for the fitness values of every individual in a population to be evaluated before proceeding to the next generation [34]. The specific multi-objective genetic algorithm implementation is written in C++ using message passing interface (MPI), described in Appendix F, for parallelization.

6.1.1 GenMOP MPI Details. GenMOP uses the basic MPI routine calls outlined in Table 11 in Appendix F. MPI_Init must be the first call in any program utilizing MPI. As arguments tt takes the command line arguments to use in the MPI environment if necessary. The server, or master processor, is brought online and then each one of the clients is initialized. The number of clients is equal to the number of processors requested by the user minus one, the master. These clients sit idle until further information is received from the master and the main genetic algorithm program is begun. When the clients for a particular generation are being evaluated individuals are dispatched to clients until they have all been evaluated. If the number of individuals exceeds the number of clients, then these individuals wait until there is a client ready to perform an additional evaluation.

The master processor waits until all individuals from a mating pool have been evaluated before beginning the ranking process in the cumulative population. Figure 22 depicts the details of the flow of the MPI parallelization used in GenMOP.

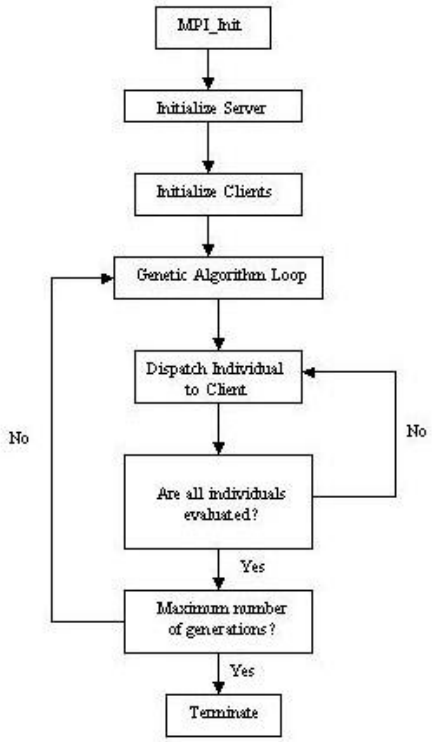

Figure 22 Flow of Message Passing Interface Utilized for Parallelization of GenMOP.

As researchers search for a solution to any given problem it is imperative that efficiency and effectiveness be their ultimate goals during the algorithm design process. Thought should additionally be given to algorithm scalability to ensure ease of future expansion and added enhancements. The remainder of this chapter discusses these performance metrics and the experiments designed to show both the efficiency and the effectiveness of GenMOP as it is applied to the QC laser simulation.

6.2 Efficiency

Communication time and data rate, or throughput, are two of the most important measures used for comparison of paralleled processes. The communication time for a

program running in parallel can be significant overhead thus increasing the overall computation time appreciably. According to [51] there are three main factors that contribute to the total communication time: startup time, t_s, per-hop time, t_h, and per-word transfer time, t_w. The delay for startup time is incurred for each message passed and involves preparation of the message, execution of the routing algorithm and establishment of an interface between the sending node and the router. The time it takes a message to travel between two nodes in the network is the per-hop time. The overall cost for communication is denoted by Equation 23 where m is the size of the message being passed. The per-hop time is excluded because within this calculation its value is negligible [51].

$$t_{comm} = t_s + t_w m \tag{23}$$

The data rate is a measure of the peak number of bytes of data that can be transferred per second and is a direct reflection on the chosen interconnection network for communication. In this specific case a fast-ethernet backplane was employed and runs at an average of 100 Mb/sec.

In addition to the two performance measurements presented above, a measurement of the speedup gained is important to understanding the benefit of parallelization. Speedup is a calculation of the performance gained by improving an algorithm. In a perfect world, the speedup calculation would be a linear model denoted by Equation 24.

$$Speedup = T_{serial}/(T_{serial}/processors) \tag{24}$$

Being that this is not a perfect world, the speedup witnessed is actually non-linear in most cases, so the time for running an algorithm in parallel is more correctly shown with Equation 25.

$$T_{parallel} = parallel\ calculation\ time + overhead \tag{25}$$

Combining Equations 24 and 25 the speedup for the optimization of the QC laser is shown in Equation 26.

$$Speedup \ = \ T_{serial}/T_{parallel} \tag{26}$$

It is also suggested in [51] that efficiency be a measure of the amount of time a processor is usefully employed. This measurement may be calculated utilizing Equation 27, where S_p is the speedup achieved in parallel and p is the number of processors employed.

$$Efficiency_p = S_p/p \tag{27}$$

Additionally, Stracuzzi, et. al, suggest that speedup for a master/slave genetic algorithm model can be calculated using Equation 28 where N is the size of the population, p is the number of processors employed, a is the time required for the master processor to perform crossover, mutation and selection while b is the time required for the slave processors to perform the fitness evaluation [74].

$$S(p, N) = \frac{a + bN}{a + \frac{bN}{p}} \tag{28}$$

6.2.1 Overview of Experimental Design. The experiments are all designed, so the metrics for parallelized programs described in Section 6.2 can be gathered. The default genetic algorithm parameters defined in Table 4 are utilized for runs. A population of twenty individuals is randomly initialized with values that remained within the range constraints defined for each variable. The mating pool size is set at ten individuals initially to allow each individual from the mating pool to be evaluated, in parallel, on a separate node. The importance of this is due to the four to seven hour approximated length of time a fitness evaluation would take. Matching individuals to nodes allows for a better use of resources by utilizing processor time and thus increasing efficiency as defined in Equation 27. In addition, no changes are made in the default mutation rate and niche radius. These parameters were initially chosen for GenMOP evaluation runs involving

different MOPs, but without a proper understanding and visualization of the search space no better numbers for these parameters are devised.

The QC laser simulation is designed, so that both the target frequency and number of layers needed to form the laser structure can be provided at runtime. Initial runs are made using a five-layer structure because previous devices fabricated with this structure have had success with lasing. Runs are made with the target frequency at both 1.5 and 3.0 terahertz. The three terahertz target is initially chosen because earlier fabricated structures have seen lasing at this frequency. The target of 1.5 terahertz is chosen because this frequency easily penetrates clothing.

To show the relative efficiency of the parallel algorithm and perform the analysis described in Equations 26, 28 and 27, GenMOP is initially run on just one node. The wall clock time is measured and recorded. This number gives a baseline to measure the parallel algorithm's performance. Using the serial run-time the communication overhead can be calculated along with the real parallel run-time. The serial run-time is also used to compute the speedup achieved through the use of parallelization.

The experiments are run on the Apsen Beowulf cluster utilizing the fast ethernet backplane for inter-node communication. Details of this hardware can be found in Appendix G. Wall clock time measurements are taken for both serial and parallel runs. Seven runs, in both serial and parallel, are performed to ensure an engineered statistical accuracy [50]. The mean, variance and standard deviation are calculated in serial and parallel. These results are graphed, compared and analyzed in Chapter 7.

6.3 Effectiveness

The American Heritage Dictionary defines effective as "having the intended or expected effect; serving the purpose". Following this definition it can be said that an effective multiobjective genetic algorithm is one which discovers one or more optimal solutions, its purpose, to the multi-objective optimization problem at hand. It handles the required solution discovery through a careful balance between exploration and exploitation within

the search space. As Eshelman, et.al., state, "the effectiveness of the GA depends upon the appropriate mix of exploration and exploitation" [24].

Exploiting the search space requires the use of accumulated information which could potentially direct the search towards enhanced solutions within the relative space [22]. A simple genetic algorithm performs exploitation through its selection operator whether successful or not. Individuals chosen to represent the population for another generation demonstrate the exploitive property of a genetic algorithm [60]. Selection for inclusion in the mating pool utilized in GenMOP through Pareto ranking as well as the niching scheme, described in more detail in Chapter 4, reveals the exploitation occurring within this algorithm.

Exploration is simply the detection of new and alternate areas in the search space. Both the crossover and mutation operators in a genetic algorithm assist in exploring additional points in the search space because creating variation within the population is the goal of these operators. Four separate types of crossover operators and three various types of mutation operators, described in Chapter 5 are utilized in GenMOP to ensure superior exploration.

It has been stated that given knowledge of the global optima for a specific problem both test suite and benchmark functions offer a means for comparing MOEA performance [82]. A summary of MOEA metrics found in the literature, taken from [17], and used to measure the performance of a MOEA can be found in Appendix H. The majority of these measurements require *a priori* knowledge of the true Pareto front and although in certain instances this can be approximated it has been shown [4] that discovery of a multiobjective problem's Pareto optimal solution set is an NP-Complete problem in itself. Without a measurement of the true Pareto front for this problem additional metrics must be employed.

To determine the effectiveness of the multi-objective genetic algorithm employed to locate solutions to this real-world problem, a measurement of the total number of non-dominated individuals found in the cumulative population is taken. This measurement, acquired from [82], is defined in Equation 29.

$$ONVG = \mid PF_{known} \mid \tag{29}$$

The overall nondominated vector generation (ONVG) measurement provides a rough assessment of the effectiveness of the algorithm's performance, but does not give insight into the percentage of possible solutions found. In order to arrive at this measurement the true pareto front must be known *a priori*. The calculation of these solutions for the real-world problem at hand is NP-Complete, as stated above, so additional methods for showing effectiveness must be explored [80].

Statistical analysis techniques, an important aspect of the testing process, are investigated. In general parametric tests such as the t-test and one-way analysis of variance (ANOVA) tests are utilized. These tests assume that the basic underlying source population is normally distributed with an additional assumption about the measurements having come from an equal-interval scale [58]. Because it is not clear through observation that the algorithm's performance metrics will be of a normal distribution non-parametric statistical testing techniques are utilized. These techniques are unique in that they make no assumptions about the defining properties of the data's population distribution. Specific non-parametric statistical tests include:

- Mann-Whitney test
- Fisher exact probability test
- Wilcoxon signed-rank test
- various forms of chi-square tests
- Kruskal-Wallis test
- Friedman test

6.3.1 Kruskal-Wallis. The Kruskal-Wallis non-parametric statistical test is chosen for data analysis purposes here. This statistical testing technique has been utilized to analyze other genetic algorithms [21] as well as numerous multiobjective evolutionary

algorithms [82]. This seems to make it a logical and fitting choice for GenMOP's statistical analysis.

The Kruskal-Wallis test is a non-parametric version of the Analysis of Variance (ANOVA) test used to compare three or more independent groups of collected data. This statistic equates to the approximation of a chi-square distribution with k-1 degrees of freedom when the sample sizes are > 5. If the sample sizes are smaller and the number of groups sampled is less than three, then a tabled value should be compared to the H statistic to determine the significance level. This statistical test is stated more formally below:

H_0: $S_1 = S_2 = ... = S_k$

H_1: $S_i \neq S_j$ for at least one set of i and j

Where H is the test statistic for Kruskal-Wallis. The significance level, α, is chosen by the experimenter and is usually a value of 0.05. The value of the test statistic is determined from Equation 30.

$$H = ((12/(n(n+1))) \sum_{i=1}^{k} (R_i^2/n_i)) - 3(n+1) \tag{30}$$

We have k independent samples of size $n_1, ..., n_k$. All of the samples are combined into one large sample and then sorted from smallest to largest. Each data point within the sample is assigned a rank with an average rank being assigned to any observation in a group of tied observations. $\bar{R}_i$, the average of the ranks of the observations in the ith sample is then found. This computed H is compared to a table of critical values based on the sample size of each group. If this H exceeds a critical value at the specified significance level, then there exists sufficient evidence to reject the null hypothesis in favor of an alternative hypothesis [72].

Kruskal-Wallis was chosen because it is important in stating effectiveness that it be shown that GenMOP generates consistently fit individuals as solutions to the multi-objective QC laser design problem. In following this statistical test's design the null hypothesis can be stated as such:

H_0: $af_1 = af_2 = ... = af_k$ where k = number of runs

H_1: $af_i \neq af_j$ for at least one set i and j

With af defining the average fitness of a GenMOP run over k runs the validity of the null hypothesis is shown with a $\alpha = 0.05$ and reported in Chapter 7.

6.3.2 Overview of Experimental Design. Experiments were designed to test the efficiency of the multiobjective GA employed to find parameter combinations that form solutions for the QC laser simulation. A five-layer structure was utilized for the reasons described previously in Section 6.2.1. Seventeen distinct frequencies, ranging from 1.0 to 6.0 THz, are investigated at 3 separate temperatures in search of feasible answers. Each of these combinations is subjected to 3 separate mutation percentages, 2, 10 and 25 percent, with populations of 25 and 50 running for two hundred and one hundred generations, respectively. An overview of these runs is shown in Table 5.

Table 5 **Genetic Algorithm Parameters for Effectiveness Experiments**

Frequency	Population/Generation	Mutation	Temperature
6.00 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
5.00 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
4.28 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
3.75 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
3.33 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
3.00 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
2.73 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
2.50 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
2.31 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
2.14 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
2.00 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
1.87 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
1.76 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
1.67 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
1.58 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
1.50 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K
1.00 THz	50/100 and 25/200	2%, 10% and 25%	10K, 60K and 80K

Thirty data runs are made for each of the combinations found in Table 5. The solutions found, space searched and overall effectiveness of GenMOP as it was applied to the QC laser problem is discussed in detail in Chapter 7.

6.4 Summary

This chapter outlined the efficiency and effectiveness experiments conducted to test GenMOP. To show efficiency the speedup, efficiency as defined in [52] and speedup for a master/slave parallel GA model are calculated. For effectiveness measurements the Kruskal-Wallis probability test is utilized and the exploration of the search space is shown. The following chapter presents the results and analysis of these experiments.

7. Results and Analysis

7.1 Introduction

The results and analysis offered here follows the experimental design outline in Chapter 6. The efficiency metrics are presented in Section 7.2 with an analysis of the speedup, efficiency as defined by Kumar [52] and the master/slave model speedup introduced in [74]. Equations 26, 27 and 28 define these concepts respectively. Section 7.3 presents the results and discussion of the effectiveness measurements obtained. The chapter is concluded with a summary in Section 7.4.

7.2 Efficiency Analysis

7.2.1 Speedup. The benefits of algorithm parallelization can be revealed through speedup calculations. Seven runs were completed in both serial and parallel for frequencies of 1.5 and 3.0 THz. The GA parameters are shown in Table 4 in Chapter 5. Eleven processors on the Apsen Beowulf system were utilized, so one could act as the master and the other ten would each correspond to an individual in the mating pool for evaluation. The wall clock time measured for each run is shown in Table 6

Table 6 **Wall Clock Time for Serial and Parallel Experiments**

		Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7
Serial	1.5THz	142.52 s	142.95 s	142.51 s	142.25 s	142.48 s	142.54 s	142.72 s
	3.0THz	142.73 s	142.96 s	142.56 s	142.47 s	142.66 s	142.86 s	142.71 s
Parallel	1.5THz	42.40 s	47.12 s	51.50 s	50.56 s	70.73 s	59.40 s	43.76 s
	3.0THz	45.80 s	42.05 s	42.54 s	41.41 s	42.49 s	42.35 s	42.23 s

For greater statistical accuracy the mean was calculated from each set of seven runs. Additionally, utilizing Equation 31, the variance of these runs was computed to show the spread of the distribution. The standard deviation, given by Equation 32, shows how far an experiment's runtime lies from the mean. A small standard deviation suggests that most of the collected run times lie somewhere close to the mean runtime. All of the previously discussed measurements are shown in Table 7. From these calculations and using Equation 26 the speedup is calculated and shown in Table 8.

$$\sigma^2 = \frac{\Sigma(X-\mu)^2}{N} \tag{31}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{32}$$

Table 7 **Mean, Median, Variance and Standard Deviation for QCL Simulation Experiments**

	Frequency	Mean	Median	Variance	Standard Deviation
Serial	1.5 THz	142.57 sec	142.52 sec	.04	.20
	3.0 THz	142.71 sec	142.71 sec	.02	.15
Parallel	1.5 THz	52.21 sec	50.56 sec	86.49	9.30
	3.0 THz	42.70 sec	42.35 sec	1.74	1.32

Table 8 **Speedup Results for QCL Simulation Experiments**

Frequency	Speedup
1.5 THz	2.73
3.0 THz	3.34

As expected, superlinear speedup is not achieved. This case is typical of most where the speedup does not exceed the number of processors used. In fact, the speedup is less than half of the number of processors utilized. This observation leads to the fact that the communication overhead is outweighing the benefits that could be realized through parallelization. It appears that the fitness evaluation time is significantly small when compared to the total elapsed time for a run. The master/slave model speedup discussed in Section 7.2.3 gives more insight.

7.2.2 Efficiency. The efficiency measurement introduced in [52] denotes the amount of time a processing element is used versus the time it spends idle. To show the efficiency of GenMOP seven runs were completed at 1.5THz utilizing 3, 5, 9 and 13 processors. The wall clock time was taken for each run, the efficiency was calculated with Equation 27 and the results are shown in Table 9.

Because the efficiency decreases as the number of processors is increased it is safe to assume that scalability could be a problem with this algorithm when small population

Table 9 **Efficiency Results for QCL Simulation Experiments**

	3 Processors	5 Processors	9 Processors	13 Processors
Efficiency	54%	51%	39%	25%

sizes are being evaluated. Again, this is due to the relatively short amount of time the fitness evaluations take in comparison to the amount of time spent doing crossover and mutation on the master processor. The communication overhead, discussed in Section 6.2, is a significant percentage of the total run time.

7.2.3 Master/Slave Model Speedup. Twenty-eight runs were completed at 1.5 THz with seven runs each for 3, 5, 9 and 13 processors. The master/slave model speedup was calculated using Equation 28. The average wall clock time for both the master and slave processors as well as the speedup calculation are shown in Table 10.

Table 10 **Speedup Results for Master/Slave Model Parallel Genetic Algorithm**

	3 Processors	5 Processors	9 Processors	13 Processors
Master Processor Time	20.07 s	15.89 s	27.30 s	30.19 s
Slave Processor Time	85.45 s	43.166 s	18.86 s	16.70 s
Master/Slave Speedup	2.99	4.93	8.22	11.034

The speedup seen for both 3 and 5 processors is very near linear. As the number of processors increases the speedup drops off slightly, but still improves the performance of the algorithm. For the smaller numbers of processors the amount of time taken by the slaves to perform fitness evaluations is significant compared to the amount of time the master takes to perform crossover and mutation. When the number of processors meets and exceeds the number of individuals that need to be evaluated the communication overhead overwhelms the benefits received through parallelization.

The current fitness evaluation for the QC laser model uses mathematical approximations to compute the required energy levels within a structure, so the evaluation time is insignificant. In future work, more accurate and computationally expensive mathematics are going to be employed. The time taken for these calculations increases significantly, so with an eye on scalability the parallelization of the algorithm is an important feature.

7.3 Effectiveness Analysis

The effectiveness of an algorithm, when applied to a specific real-world problem, is often judged on the amount and quality of solutions found. Initially, approximately 10,000 runs were made with the QC laser model and GenMOP algorithm integrated. Table 5 decomposes the frequencies, populations, generations, mutation rates and temperatures used. The first objective was reaching the target frequency specified within a range of +/- 0.0005 eV. The second objective was creating an energy level spacing, between levels E_2 and E_1 that corresponded to the interface phonon energy with a tolerance of +/- 0.002 eV.

For the first 10,000 runs solutions were found at all frequencies for objective two. Figures 23, 24 and 25 denote the fitness landscapes for three separate frequencies with respect to objective two. As shown, the landscapes in all three figures are particularly rugged and solutions appear to be located in generally the same areas.

For Figure 23 the data was captured over 100 generations with a population of 50 individuals, a mutation rate of 0.25 and at a temperature of 80 K. The thicknesses for layers 1 and 2 in the QC laser structure were graphed with respect to their objective two fitness value.

Figure 24 graphically represents the thicknesses of layers four and five of a structure lasing at 2.5 THz with respect to objective two. The data for this figure was collected over 100 generations with a population of 50 individuals, a mutation rate of 0.1 and at a temperature of 80 K.

Figure 25 graphically represents the thicknesses of layers four and five of a structure lasing at 6.0THz with respect to objective two. The data for this figure was collected over 100 generations with a population of 50 individuals, a mutation rate of 0.25 and at a temperature of 60 K.

The ruggedness of the terrain and the number of peaks present in the search space create difficulties for GenMOP during execution. During an entire GenMOP run 5000 individuals are evaluated by the algorithm. At a frequency of 1.67 THz, over 100 generations with 50 individuals evaluated per generation at a mutation rate of 0.25 and a temperature

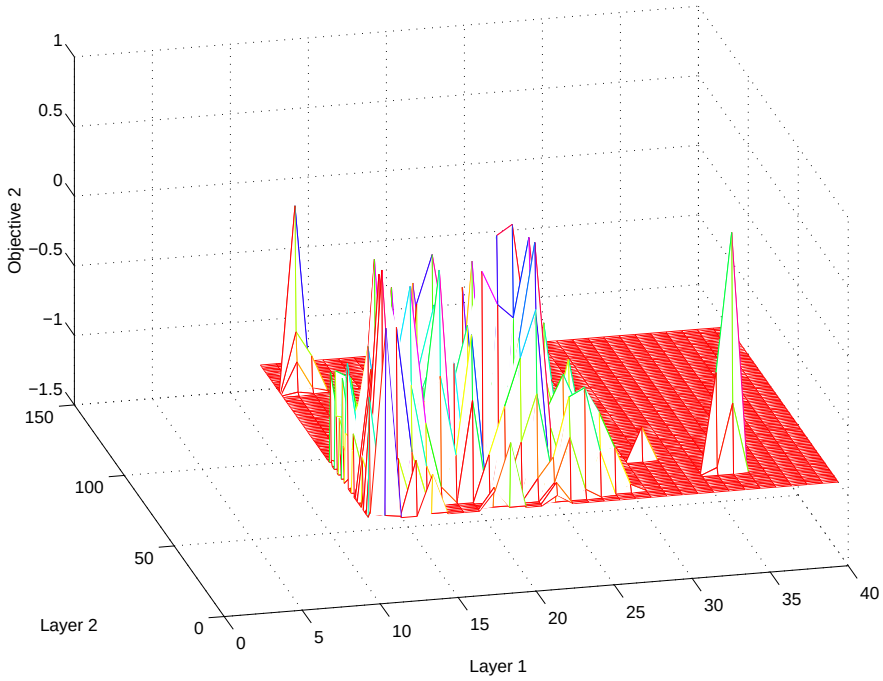

Figure 23 Fitness Landscape Showing Objective Value 2 with Thicknesses for Layers 1 and 2 at 1.67 THz

of 80 K the average number of solutions found for objective two over thirty runs was 3104. That is, out of 150,000 individuals evaluated only 93,143 were solutions. This equates to 62% of the individuals being evaluated deemed as solutions for objective two.

Over the initial 10,000 runs no parameter combinations were found that provided solutions to objective one. From the discussion above it was important to ensure that GenMOP was effectively exploring the search space and not becoming caught in some local minimum. Figures 26, 27 and 28 correspond to the landscapes produced in Figures 23, 24 and 25 described previously.

Figure 26 shows the values evaluated for the first three layers in the QC laser structure. These layers can be varied from 5 to 150 Angstroms. If this was a deterministic algorithm and every point was evaluated, then the figure would represent a solid cube. Each x in the figure denotes one combination of layer thicknesses that were evaluated. Figures 27 and 28 show the values evaluated for layers four and five and the bias. Additional graphs can be found in Appendix E.

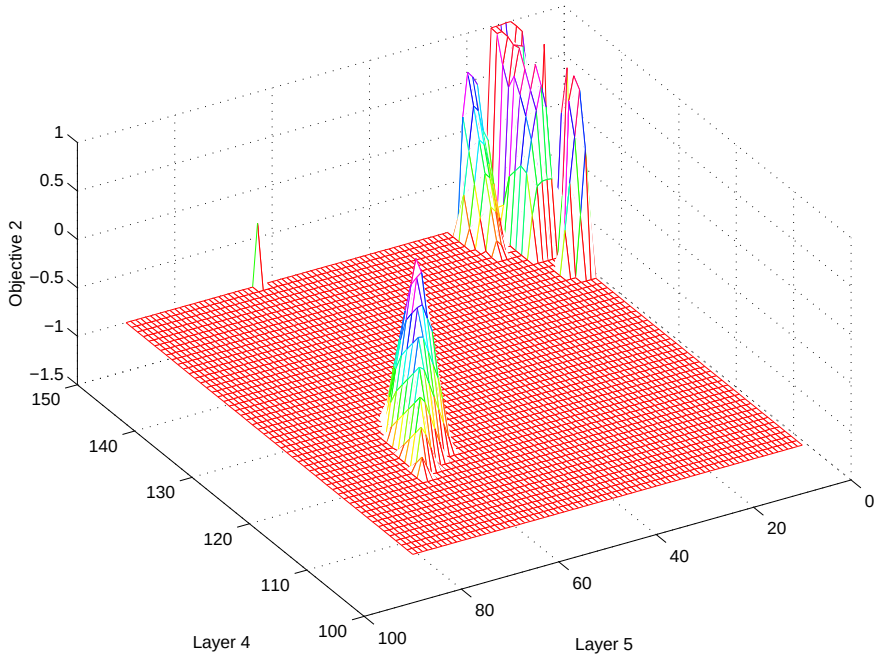

Figure 24 Fitness Landscape Showing Objective Value 2 with Thicknesses for Layers 4 and 5 at 2.5 THz

Because no laser parameter combinations that produced solutions for objective one were found and the algorithm appears to be experiencing good exploration the constraints were relaxed and additional runs completed. The original constraint meant meeting the target frequency with a tolerance of +/- 0.0005 eV. This tolerance was relaxed to +/- 0.001 eV. Runs at both 1.5 and 3.0 THz were completed with populations of 25 and 50 running 200 and 100 generations, respectively. Three different mutation rates were employed at a temperature of 10 K.

These additional runs did not discover any parameter combinations that produced any results that created solutions for objective one. The constraints were relaxed a second time for objective one. The original target frequency tolerance was changed from +/- 0.0005 eV to +/- 0.005 eV and runs were submitted at 1.5 and 3.0 THz. Again, these runs did not produce any solutions for objective one.

Directions for future research to find solutions that equate to the target frequency for the muiltiobjective QC laser optimization problem are given in Section 8.3.

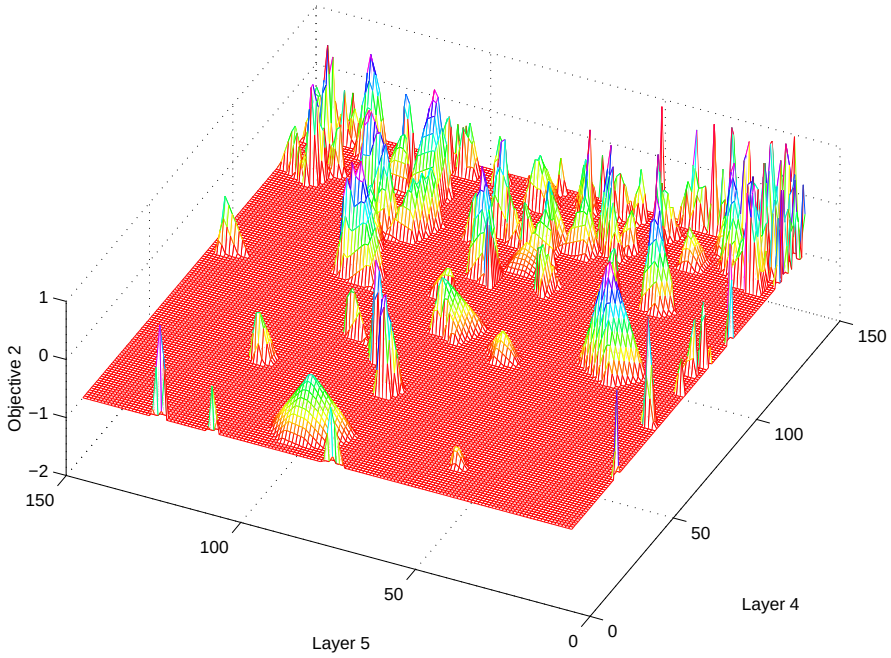

Figure 25 Fitness Landscape Showing Objective Value 2 with Thicknesses for Layers 4 and 5 at 6.0 THz

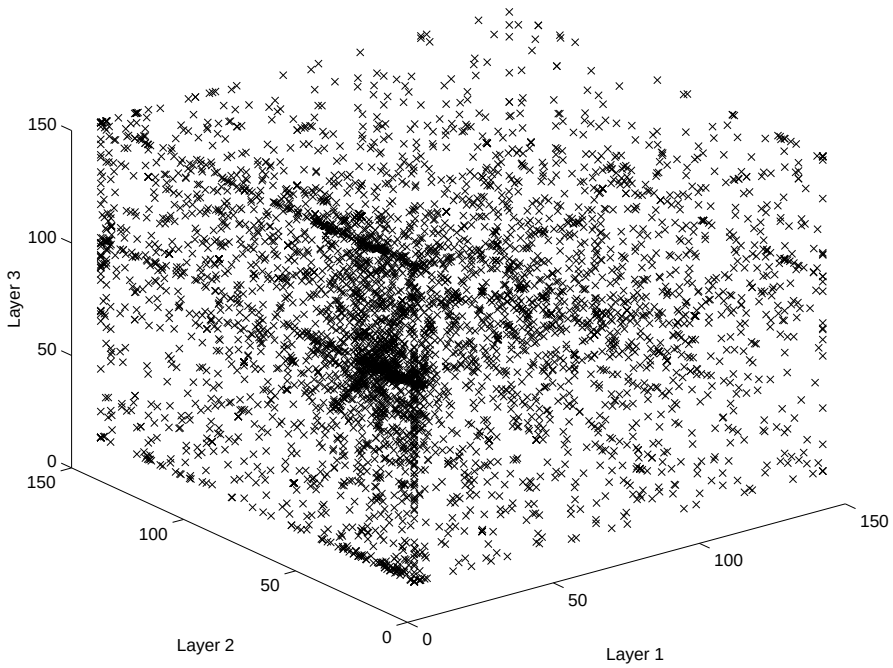

Figure 26 Values Searched for Layers 1, 2 and 3 at 1.67 THz

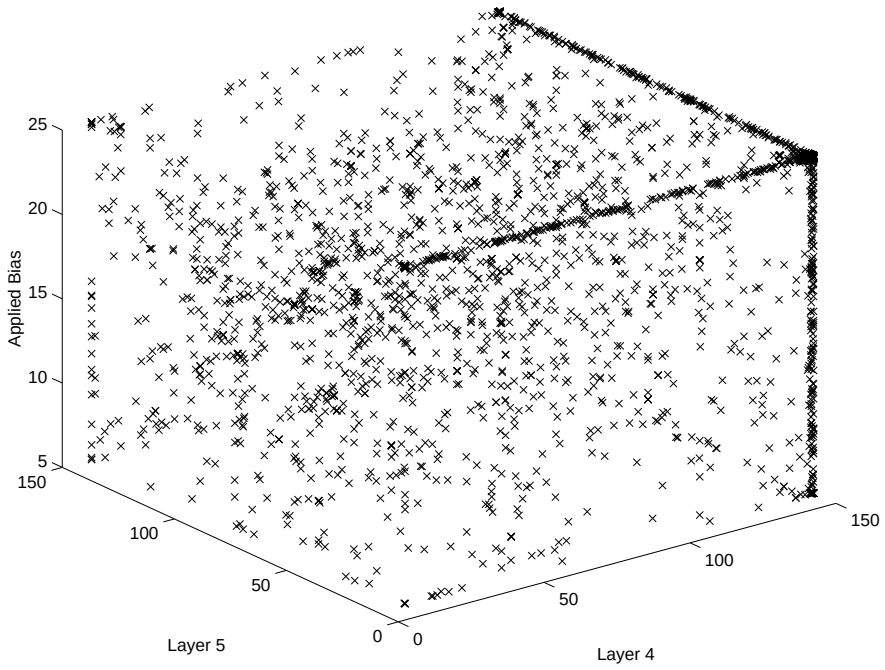

Figure 27 Values Searched for Layers 4 and 5 and Bias at 2.5 THz

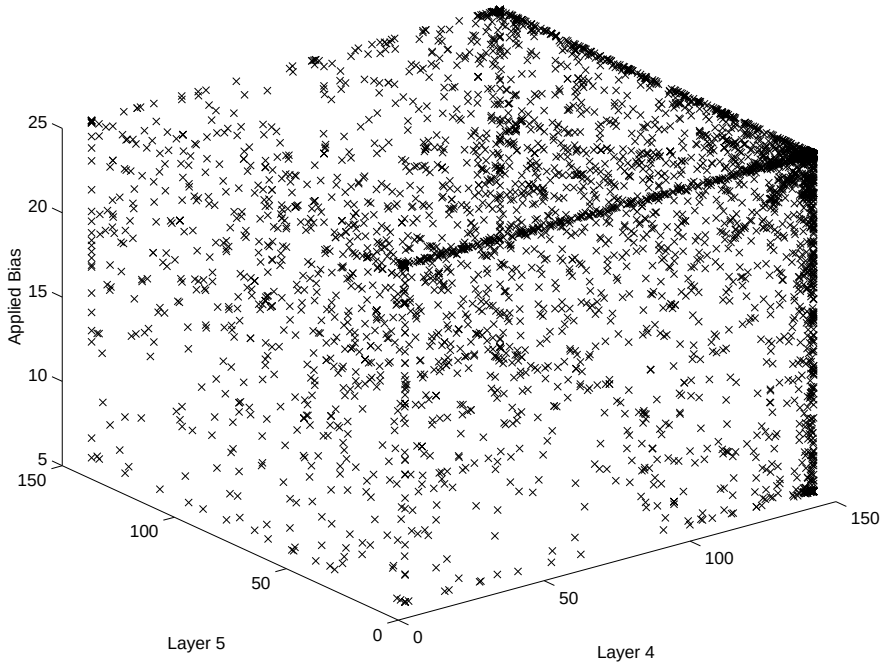

Figure 28 Values Searched for Layers 4 and 5 and Bias at 6.0 THz

7.3.1 Kruskal-Wallis Statistical Test. Twenty-one simulations were run at 1.5 THz with a population of 50 individuals for 100 generations utilizing a mutation rate of

2% at a temperature of 60 K. The number of parameter combinations found creating a solution for objective two was counted. Each of these solutions was ranked and the average rank computed. Equation 30 from Chapter 6 was used to measure GenMOP's ability to consistently create fit individuals. The test statistic, H, was calculated to be 10.8. For twenty-one runs the critical value for a significance level of .05 is 11.6 [58]. Our critical value falls below this, so we can accept the hypothesis as stated in Chapter 6 where,

$H_0 : af_1 = af_2 = ... = af_k$, where k = number of runs.

This test shows that GenMOP is able to produce reliably fit individuals as solutions for objective two.

7.4 Summary

This chapter gave an analysis of the efficiency and effectiveness results furnished from the tests conducted with GenMOP. The speedup, efficiency as introduced by [52] and master/slave model speedup defined by [74] were all discussed in Section 7.2. The effectiveness of the algorithm with respect to its exploration of the search space and success in finding parameter variations that meet the constraints for objective two was also discussed.

8. Conclusions and Recommendations

8.1 Introduction

This research provids the background knowledge necessary to appreciate the quantum cascade laser structure and operation. It begins, in Chapter 2, with a discussion of the atom, continus with laser fundamentals, semiconductors and quantum wells and concludes with information on generic QC lasers. In addition to this MOPs are discussed along with the algorithms most commonly applied and found in the literature. These include simulated annealing, Tabu search, evolutionary algorithms such as the artificial immune system and genetic algorithms, and multiobjective evolutionary algorithms. Because GenMOP, the MOEA employed in this research, has parallel capabilities a discussion of parallelization techniques for GAs is given.

Intricate details pertaining to the QC laser modelled are given in Chapter 4. These included the GaAs/AlGaAs material, wavefunctions, gain calculations and scattering effects. The specific algorithm designed for use with the QC laser software model is defined in Chapter 5. The implementation details for the concepts of crossover, mutation and equivalence class sharing utilized in the General Multiobjective Parallel Genetic Algorithm (GenMOP) are discussed. Following this the experiments designed are outlined in Chapter 6. The results of these experiments are presented in Chapter 7. The efficiency and effectiveness are discussed and an analysis is offered.

This chapter draws conclusions about the research in Section 8.2 and offers directions for future work in Section 8.3.

8.2 Conclusions

Conclusions can be drawn about both the algorithm efficiency and it's effectiveness when utilized to solve a multiobjective QC laser problem.

8.2.1 Algorithm Efficiency. The efficiency of an algorithm is often measured by the speedup produced through parallelization. The speedup achieved is a calculation of the serial processing time versus the parallel processing time which takes communication

overhead into account. The relatively small evaluation time required for an individual due to the mathematical approximations utilized in the QC laser model created a situation where the communication overhead was a large percentage of the overall run time. This meant that for small populations of individuals at, or exceeding the number of processors utilized the speedup was less than linear. In addition, the efficiency, as defined by Kumar, dropped below 50% for runs where the number of processors utilized was equal to, or greater than, the number of individuals being evaluated.

Although this was the case the algorithm lends itself to scalability as the population increases. The parallelization is also a key ingredient when, in future work the mathematical approximations are exchanged for more exact and computationally expensive methods.

8.2.2 Algorithm Effectiveness. An effective multiobjective genetic algorithm is initially described in Chapter 6 as one that would find one, or more optimal solutions to the MOP at hand. Added to this description to show effectiveness is the exploration the GA experiences during execution. The initial 10,000 runs produced numerous parameter combinations that met the constraints for objective two, but none that satisfied those for objective one. To ensure the algorithm is effectively searching the rugged landscape each point evaluated by the GA is graphed. These graphs are shown in Chapter 7 and Appendix E. GenMOP experiences good exploration, so the constraints on objective one are relaxed and new runs completed. These runs produced no solutions, so again the constraints are relaxed and runs are completed. This second relaxation produced no solutions as well. Thoughts on this and directions for future work are discussed in the next couple sections.

8.3 Future Work

The lack of solutions that equate to the target frequency resulting in the absence of any optimal solutions to this MOP results in many possibilities for future research. Even though the constraints have been relaxed twice there is a possibility that further relaxation would result in GenMOP finding solutions. If this approach is taken and solutions are found, then a local search could be employed with the GA to exploit the area where those

solutions are found. Additionally, once a solution has been found the parameter ranges can be constrained and the GA itself re-employed to search in this area.

The GA may also benefit from more individuals in the population and possibly more generations. The Aspen Beowulf cluster was the restricting factor here because it has a limited number of nodes. The Major Shared Resource Center (MSRC) maintains much larger computer systems than the Air Force Institute of Technology (AFIT), so it should be utilized. This would allow the population to be increased and the parallelization of the program exploited.

Once preliminary solutions are gathered additional variables may be added to the MOP. These include temperature variation during execution and donor densities for each layer. The known solutions may be used to initialize a population, so the GA will be less likely to become trapped in a local minima due to the rugged landscape.

Additionally, the mathematical calculations in the QC laser model software can be changed from approximations to the full computation. This increases the accuracy and strengthens the discovered solutions. It also increases the fitness evaluation time and makes better use of the parallelization aspect of GenMOP. Additional objectives such as gain maximization may be added and the solutions found for objectives one and two may again be used to create an initial population.

8.4 Summary

This chapter offered conclusions on the algorithm's efficiency and effectiveness evaluations given in Chapter 7. The efficiency of the algorithm was shown in regards to its parallelization while the effectiveness was due to the exploitation of the search space. Directions for future work were discussed. These included QC model mathematical exactness, adding variables, relaxing constraints, adding objectives and increasing population sizes by utilizing the MSRC.

Appendix A. Multiobjective Optimization Techniques

A.1 Simulated Annealing

Simulated annealing is a local search method, that falls in the class of threshold algorithms, which escapes the difficulty many searches have of becoming stuck in a local optima by moving to non-improving states with some probability. The probability of accepting j given that we are at state i is described mathematically below [1].

$$P_{c_k}\{acceptj\} = \begin{cases} 1 & \text{if f(j)} \leq \text{f(i)}, \\ exp\left(\frac{f(i)-f(j)}{c_k}\right) & \text{if f(j)} \geq \text{f(i)}. \end{cases} \tag{33}$$

A neighborhood is defined for each state, s, using a predetermined neighborhood function. A particular neighbor, s', is chosen from the neighborhood, *N(s)*, so *s' ∈ N(s)*. If an *s'* is chosen such that *f(s) ≥ f(s')*, where f represents the cost function, then s' will replace s. If the previously mentioned condition does not hold, then *s'* will replace *s* with a probability computed from the equation above [37].

At Arizona State University in the early '90s a simulated annealing technique was applied to the MOP of designing intelligent structures. Aditi Chattopadhyay and Charles Seeley were interested in the synthesis of structures/controls and the actuator location problem, which requires the optimization of competing objectives such as vibration reduction, dissipated energy, power and a performance index. Using a simulated annealing algorithm they were able to meet all of the problem imposed constraints while finding solutions that resulted in significant improvements for all their objectives [14].

The technique of simulated annealing is adapted to solve MOPs such as energy minimization in physical systems and objective function minimization in structural systems [8], the loading of nuclear fuel assemblies [79] and many others discussed in [20].

A.2 Tabu Search

Tabu search (TS), developed by Fred Glover in 1986, is an extension of the local searching technique used to overcome the problem of being trapped in a local optima [31].

Much like a local search, Tabu search uses a search space, or set of solutions, cost function and a neighborhood function to perform its search. The secret behind Tabu search is that, besides storing the value of the best solution found thus far, it stores information about previously visited solutions in a *tabu list*. This list keeps the search from cycling or becoming trapped in an infinite loop [37]. The list also serves to allow the search to choose a solution that is less than optimal in order to avoid a path that has already been visited. The choice of a less than optimal solution will help tabu search discover the global minima rather than becoming fixed on a local minima.

Although MOPs solved using tabu search utilize the basic principles, described for use with a single-objective optimization problem above, they often differ in the details. Michael Hansen developed a tabu search to find solutions to a problem modeled after the NP-complete knapsack problem. For his problem there existed a set of fifty locations, each with an equal probability of being chosen as a member of a solution which establishes a chain of gas stations under a particular budget. To find optimal solutions a set of three objectives is evaluated. Both the short and long term profits must be maximized while the negative impact on the environment is minimized. Hansen's multiobjective tabu search (MOTS) attempts to find solutions which cover the entire Pareto frontier by choosing points that optimize by diverging from already existent solutions.

When the MOTS algorithm begins a number of random, feasible solutions are found to replace the current solutions and the tabu list (TL) that each solution maintains is emptied. The set containing all non-dominated solutions (ND) is emptied, the range equalization factors (π) are set to a unit vector and an iteration counter is set to zero. After these initialization steps are completed a loop is entered which allows each solution to move towards a neighboring solution one step at a time until the predetermined stopping criteria is met and the algorithm terminates with all the encounter non-dominated solutions in the set ND.

To ensure solutions entered into ND are spread equidistantly over the Pareto frontier each solution maintains a weight vector, λ. Elements in λ correspond to the Manhattan distance of each point from the current solution. Points lying closer to the current solution carry more weight to ensure the solution will shift away from these points. A solution to

replace the current is chosen which is shown to be the best feasible solution within the neighborhood that does not interfere with the tabu list. To assist the algorithm in choosing feasible solutions a minimum acceptance level is set for the objectives. Additionally, saturation levels are established that once reached denote the objective needs no further optimization. The use of both the minimum and saturation levels may increase the speed and effectiveness of the algorithm. A problem with a known solution is unique in that it allows an algorithm's effectiveness to be perfectly tested by comparison to a reference of the known, optimal solution.

Although Hansen found success using TS to solve the knapsack problem this method does not seem to be as prevalent in the literature when it comes to real-world MOPs [75].

A.3 Artificial Immune Systems

The biological immune system (BIS) is a sophisticated information processor. It is a highly adaptive system with the ability to learn from its environment and maintain a history of past encounters. The cells that make up the BIS are distributed throughout the body, working in parallel without a central controlling mechanism. The field of study that incorporates an artificial immune system (AIS) into combinatorial problem solutions derives its inspiration from the biological world.

The artificial immune system (AIS) uses the biological immune system (BIS) as a model for operation. It takes the basic ideas such as detection, learning and history maintenance, that make the BIS so successful and applies them to optimization problems [25].

Appendix B. Pareto-based Multiobjective Evolutionary Algorithms

The following sections, taken from [47] describe various Pareto-based MOEAs.

B.1 Golberg's Pareto Ranking

Goldberg suggested moving the population toward PF_{true} by using Pareto nondominated points and selection [32]. To get the population for the next generation, the nondominated Pareto Fronts are determined and are ranked based on best solution set to the worst. Once the number of individuals ranked matches the number of individuals needed for the next generation, the process is terminated.

B.2 Multi-objective Genetic Algorithm

Fonseca uses a ranking approach different from Goldberg. He ranks the points based on how many other points dominate them. His first rank is identical to Goldberg's first. But the rest of the ranks are dependent upon how dense the population is in front of the point. The multi-objective genetic algorithm (MOGA) uses a niche-formation method in order to diversify the population [27]. Since MOGA niching is done in the objective space, individuals that map to the same objective value will only have one member kept in the population and all others removed. This is a disadvantage of the algorithm [**?**].

B.3 Nondominated Sorting Genetic Algorithm

This method, presented in [73], ranks members based on the size of the population when they are nondominated. This results in the better members getting the higher fitness scores. Selection is done using stochastic remainder proportionate selection to ensure copies are distributed properly. An offshoot of this approach, Nondominated Sorting Genetic Algorithm II (NSGA-II), is more efficient and uses elitism. This method tends to perform worse than MOGA in tests and may be due to the sharing factor being improperly set [**?**].

B.4 Niched-Pareto Genetic Algorithm

This method employs an interesting form of tournament selection called Pareto domination tournaments. two members of the population are chosen at random and they are each compared to a subset of the population. If one is nondominated and the other is not, then the nondominated one is selected. If there is a tie (both are either dominated, or nondominated), then fitness sharing decides the tournament results [41].

B.5 Strength Pareto Evolutionary Algorithm

This method attempts to integrate different MOEAs. First introduced in [93], the algorithm uses a strength variable that is similar to the MOGA ranking system. Each member of the population is assigned a fitness value according to the strengths of all nondominated solutions that dominate it. Diversity is maintained through the use of a clustering technique called the "average linkage method".

A revision of this method, called Strength Pareto Evolutionary Algorithm 2 (SPEA) [90], adjusts slightly the fitness strategy and uses nearest neighbor techniques for clustering. In addition, archiving mechanism enhancements allow for the preservation of boundary solutions that are missed with SPEA.

B.6 Multi-Objective Messy Genetic Algorithm

This method extends the messy genetic algorithm [45] to solve multiobjective problems. The Multi-objective Messy Genetic Algorithm (MOMGA) [80] is an explicit building block GA that produces all building blocks of a user specified size. the algorithm has three phases: Initialization, Primordial and Juxtapositional. For explicit details of how this algorithm functions see [47]. A disadvantage of this algorithm is the exponential growth of the population as the building block size grows.

B.7 Multiobjective Hierarchical Bayesian Optimization Algorithm

This explicit building block method combines the multiobjective selection scheme of NSGA-II with the Hierarchical Bayesian Optimization Algorithm (hBOA) [64]. The

Multiobjective Hierarchical Bayesian Optimization Algorithm (mhBOA) [46] is a linkage learning algorithm that attempts to define tight and loose linkages to building blocks in the chromosome. This method uses a Bayesian network (a conditional probabilistic model) to guide the search toward a solution. A disadvantage of this algorithm is the time it takes to generate results with just a small number of linkages tested.

B.8 Pareto Archived Evolution Strategy

This method, formulated by Knowles and Corne [49], uses a (1+1) evolution strategy, where each parent generates one offspring. the method uses an archive of nondominated solutions to compare with individuals in the current population. For diversity, the algorithm generates a grid overlaid on the search space and counts the number of solutions in each grid. A disadvantage of this method is its performance on disconnected Pareto Fronts.

B.9 Pareto-based Selection

These are approaches that do not use niching, sharing, or crowding. In order to maintain diversity, other methods need to be devised. Several different approaches are described in [**?**].

B.10 Pareto Deme-based Selection

This approach applies Pareto ranking on many small subpopulations. This approach fits nicely into the parallel processing paradigm. A new method must be created in order to determine which nondominated subpopulation members are also globally nondominated.

B.11 Pareto Elitist-based Selection

These approaches take the best n individuals from one generation and propagate them to the next. After that the rest of the population is filled using some other method. Large selection pressure for this approach can cause premature convergence [**?**].

Appendix C. Load Balancing

C.1 Static Load Balancing

C.1.1 Recursive Coordinate Bisection. Recursive coordinate bisection is one of the most simplistic partitioning algorithms. It partitions a graph according to the coordinates of its vertices by fist determining which coordinate direction is the longest. The points within the domain are then arranged in order according to this coordinate and divided, so that an even number lies on each side of the partition. This process is continued, recursively, until the intended number of partitions is reached. Disconnection among the subdomains may occur if the points were not evenly spread across the domain because recursive coordinate bisection does not make use of any connectivity information [78].

C.1.2 Recursive Graph Bisection. Unlike recursive coordinate bisection which utilizes coordinate distance to partition the domain, recursive graph bisection exploits graph distance to perform this task. This method is used to reduce the number of grid edges that cross partition boundaries. Initially, the two vertices in the graph that are the maximum distance apart are found. One of these vertices is chosen as the root and all other vertices in the graph are ordered according to their distance from the root. The set of vertices is divided in half, creating two subdomains where it is guaranteed that at least the subdomain containing the root is connected.

C.1.3 Recursive Spectral Bisection. Recursive spectral bisection is a recursive bisection method used to partition unstructured problems. In order to perform recursive spectral bisection the steps below must be followed:

1. Compute the Laplacian matrix, L, for the graph.

$$L = -D + A, where \tag{34}$$

$$A_{ij} = \begin{cases} 1 & \text{if edge}(v_i, v_j) \in Graph, \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

$$D_i = degree(v_i) \tag{36}$$

2. Locate the smallest non-zero eigenvalue, λ_f, and corresponding eigenvector, x_f.
3. Determine the median value of the entries in x_f.
4. Values greater than the median form one subdomain, remaining values form a second subdomain.
5. Recurse until enough partitions have been created.

Although balancing communication with workload by finding a suitable partition is an NP-complete problem, recursive spectral bisection is an algorithm that provides a good approximation for a number of real world problems [6].

C.1.4 Scattered Decomposition (modular mapping). When faced with an highly irregular domain, scattered decomposition is the method of choice for decomposing a problem. This technique partitions the data into a number of rectangular clusters, r, so that there are more clusters than processors [52]. A balance between the number of clusters needed and the tolerable communication overhead must be found for an ideal load balance because amount of communication required increases with r [70]. A pictorial example of scatter decomposition can be seen in Figure 29.

C.2 Dynamic Load Balancing

Dynamic load balancing occurs during process execution which incurs an additional overhead, but is often more effective than static load balancing. Dynamic load balancing algorithms use current load information to make decisions about work distribution. Although most dynamic load balancing algorithms are specific to the application domain there are four main steps that they all share in common [88]:

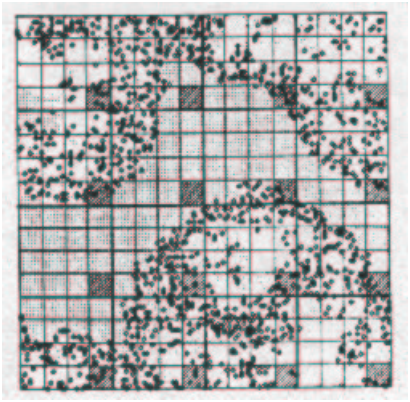

Figure 29 An Example of Scatter Decomposition

1) Load Monitoring

2) Synchronization

3) Rebalancing Criteria

4) Job Migration

The main advantage of dynamic load balancing over static is its flexibility which allows for adaptation when unforeseen requirements arise at run-time.

C.2.1 Centralized Dynamic Load Balancing. Centralized dynamic load balancing, also known as the pool-based method (PBM), uses a fixed processor as the master which holds all the available work. When a slave processor has finished performing its tasks and is idle, then it requests additional work from the master processor. This process is shown in Figure 30.

C.2.2 Distributed Dynamic Load Balancing. Distributed dynamic load balancing (DLB), also known as a peer-based method, initially distributes the work evenly among all the processors. A processor that is idle, having completed its work, will select another processor to request work from through one of four methods, asynchronous round robin, global round robin, random polling or nearest neighbor.

For the asynchronous round robin scheme of distributed DLB each processor possesses their own label and a target. This target is the label of the processor which they request

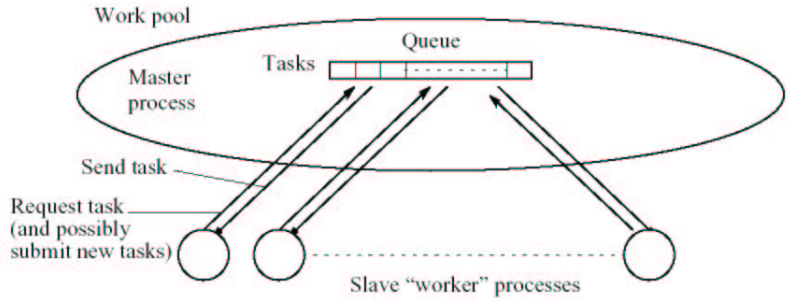

Figure 30 How Centralized Dynamic Load Balancing Works

work from if they are idle. The initial value of the target is *(their label + 1) modulo p*, the number of processors. Each time a processor requests additional work their target is incremented by one modulo p. If the processor requests from a target that does not have extra work, then it follows the incrementing process and continues requesting.

Global round robin is much the same as asynchronous round robin, but there is only one *global* target maintained. When a processor is idle and wishes to request work, it first requests the target value and then sends a request for additional work to this target. While work is being requested from the target some synchronization routine, such as a semaphore, is employed, so that no other processor may access the target. After each request the target is incremented *modulo p*.

If a processor becomes idle and the random polling technique is being employed, then a target is selected at random to request work. Each processor is selected randomly with an even probability to ensure that work request are distributed uniformly [51].

There are many variations of the nearest neighbor load balancing method, such as dimension exchange, diffusion and their variants. These algorithms allow processors to make work requests based on the workload in their neighborhood.

```
#:=====================================
#: File: menon035.inp
#: Created: May 15, 2002 by J.A. Gagnon
#: Modified: July 24,2003 by Alexi Girgis
#: Input file for Terahertz Laser Calculations
#:=====================================
#: This input file MUST have '#:' as the firs two characters in comment lines !!!
#: Otherwise, if a '#' is missing for example, the program will read in garbage!!!
#: So check the output to see that the numbers are reasonable. () are for recommended
#: values or to indicate what a particular choice enables.
#:=====================================
#:Display Format (FILE / SCREEN / BOTH / NONE)
#:=====================================

              NONE
#:=========================================
#:System Properties
#:=========================================
#:.....
#: Bias in kV/cm
#:.....

              19.6
#:.....
#: Temperature in Kelvin
#:.....

              10
#:.....
#: BC type
#: Infinite QW BC = 0
```

#: Trav. Wave BC = 1

#:.....

1

#:.....

#: Number of layers

#:.....

5

#:.....

#: Number of subbands

#:.....

3

#:.....

#: Subband Populations ($10^1$1) ($1/cm^2$)

#: 1, 2, 3 . . Number of subbands

#:.....

0.7

#:.....

0.9

#:.....

1.1

#:.....

#:

#:=======================================

#:Layer Properties

#:=======================================

#:conc is the impurity concentration. If the material is binary

#:this entry must be 0.

#:.....

#:Layer 1

```
#:.....
#:Width — nelem — Ddensity — Denergy — metallic(0/1) —
#:.....
        40        14          1.0e16          0.005          0
#:.....
#:Material — conc — outer gausselem — inner gausselem —
#:.....
       AlGaAs       0.3              14                  14
#:.....
#:
#:.....
#:Layer 2
#:.....
#:Width — nelem — Ddensity — Denergy — metallic(0/1) —
#:.....
       72        14          0.0          0.005          0
#:.....
#:Material — conc — outer gausselem — inner gausselem —
#:.....
       GaAs       0.0              14              14
#:.....
#:
#:.....
#:Layer 3
#:.....
#:Width — nelem — Ddensity — Denergy — metallic(0/1) —
#:.....
       20        14           1.0e16           0.005           0
#:.....
```

```
#:Material — conc — outer gausselem — inner gausselem —
#:.....
      AlGaAs      0.3             14              14
#:.....
#:
#:.....
#:Layer 4
#:.....
#:Width — nelem — Ddensity — Denergy — metallic(0/1) —
#:.....
      62      14          0.0          0.005             0
#:.....
#:Material — conc — outer gausselem — inner gausselem —
#:.....
      GaAs      0.0         14             14
#:.....
#:
#:.....
#:Layer 5
#:.....
#:Width — nelem — Ddensity — Denergy — metallic(0/1) —
#:.....
      96       14           1.0e16           0.005           0
#:.....
#:Material — conc — outer gausselem — inner gausselem —
#:.....
      AlGaAs       0.3              14              14
#:.....
#:
```

```
#:=========================
#:Output Options
#:=========================
#:.....................................................................
#:Plot potential? (0/1)
#:.....................................................................
                1
#:.....................................................................
#:Plot energies? (0/1)
#:.....................................................................
                1
#:.....................................................................
#:Plot wavefunctions? (0/1)
#:.....................................................................
                1
#:.....................................................................
#:============================
#:Self Consistency Options
#:============================
#:.....................................................................
#:Enabled? (0/1)
#:.....................................................................
                0
#:.....................................................................
#:Inner Tolerance (0.000001) Outer Tolerance (0.00001)
#:.....................................................................
                0.00001         0.00001
#:.....................................................................
#:==============================
```

```
#:E-E Scattering Options
#:==============================
#:.....
#:Enabled? (0/1)
#:.....
                0
#:.....
#:Max. Iterations Calls per Iteration
#:.....
                50          200
#:.....
```

Figure 31 represents the thicknesses of layers four and five of a structure lasing at 1.0THz with respect to objective two. The data for this figure was collected over 200 generations with a population of 25 individuals, a mutation rate of 0.1 and at a temperature of 10 K.

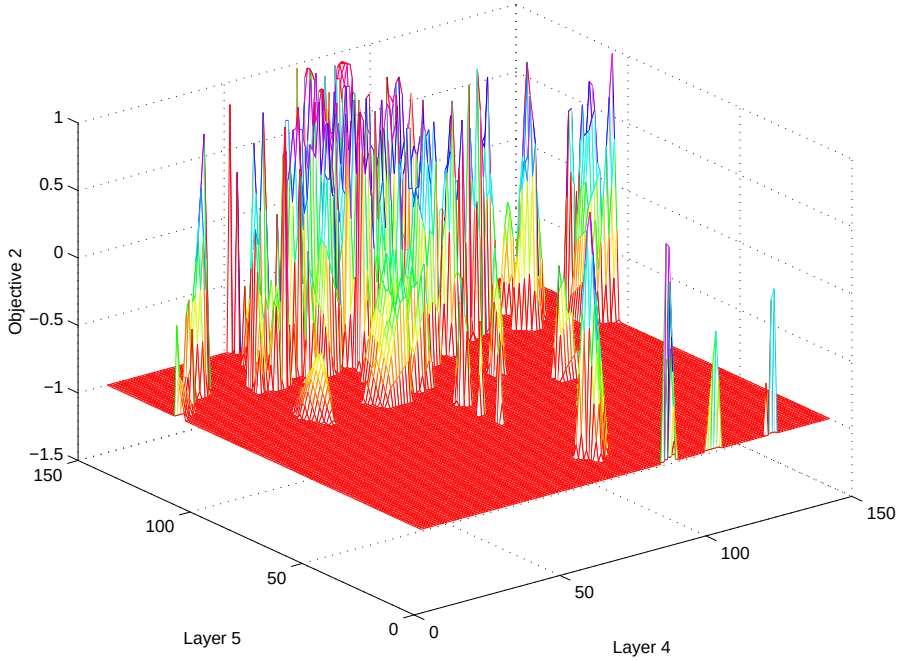

Figure 31 Fitness Landscape Showing Objective Value 2 with Thicknesses for Layers 4 and 5 at 1.0 THz

Figure 32 shows the values of layers 4 and 5 and the bias evaluated at 1.0 THz over 200 generations with a population of 25 individuals, a mutation rate of 0.1 and at a temperature of 10 K.

Figure 33 represents the thicknesses of layers four and five of a structure lasing at 1.5 THz with respect to objective two. The data for this figure was collected over 200 generations with a population of 25 individuals, a mutation rate of 0.1 and at a temperature of 10 K.

Figure 34 shows the values of layers 4 and 5 and the bias evaluated at 1.5 THz over 200 generations with a population of 25 individuals, a mutation rate of .1 and at a temperature of 10 K.

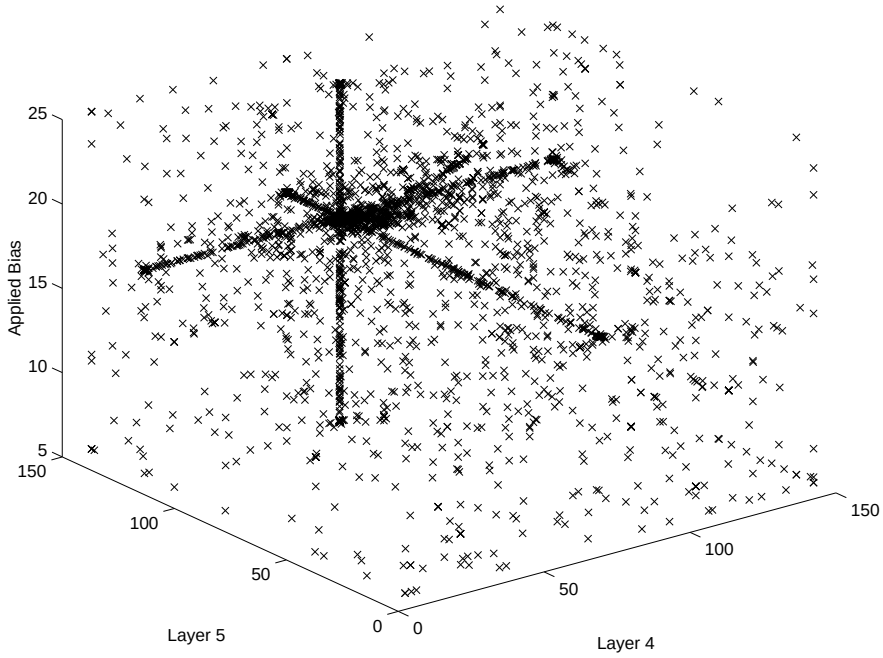

Figure 32 Values Searched for Layers 4 and 5 and Bias 1.0 THz

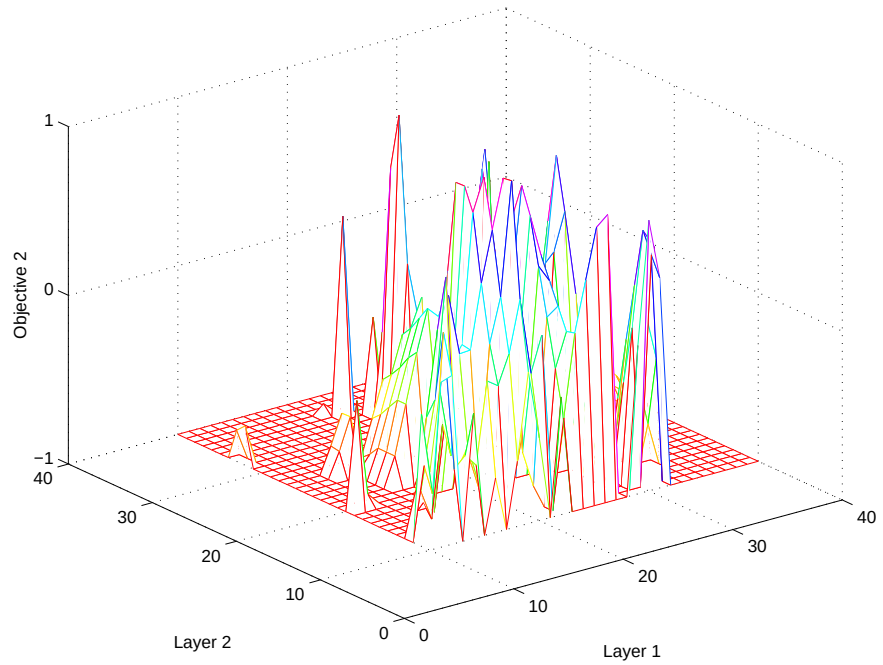

Figure 33 Fitness Landscape Showing Objective Value 2 with Thicknesses for Layers 1 and 2 at 1.5 THz

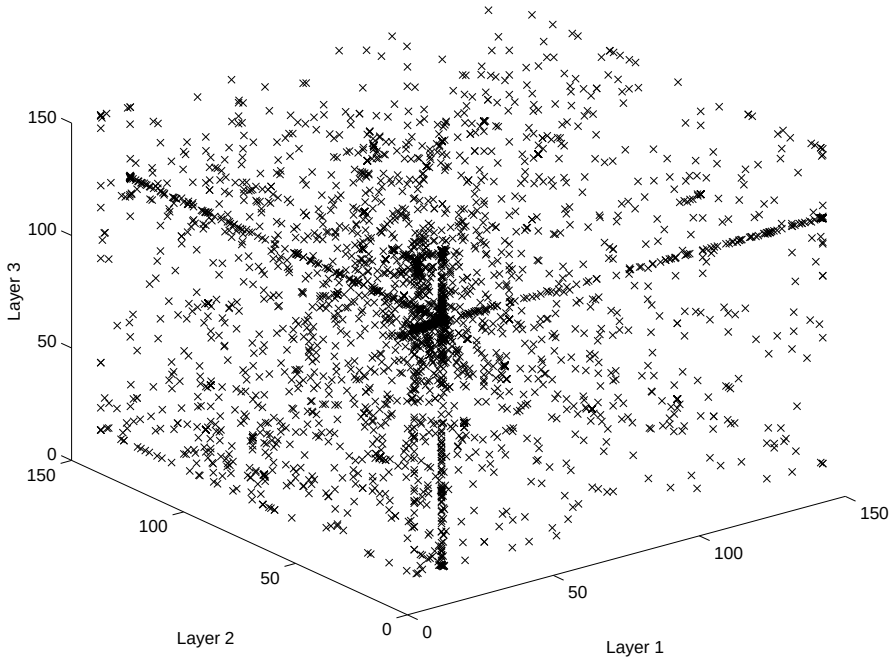

Figure 34 Values Searched for Layers 1,2 and 3 at 1.5 THz

Figure 35 represents the thicknesses of layers four and five of a structure lasing at 1.5 THz with respect to objective two. The data for this figure was collected over 200 generations with a population of 25 individuals, a mutation rate of 0.02 and at a temperature of 60 K.

Figure 36 shows the values of layers 4 and 5 and the bias evaluated at 5.0 THz over 200 generations with a population of 25 individuals, a mutation rate of 0.02 and at a temperature of 60 K.

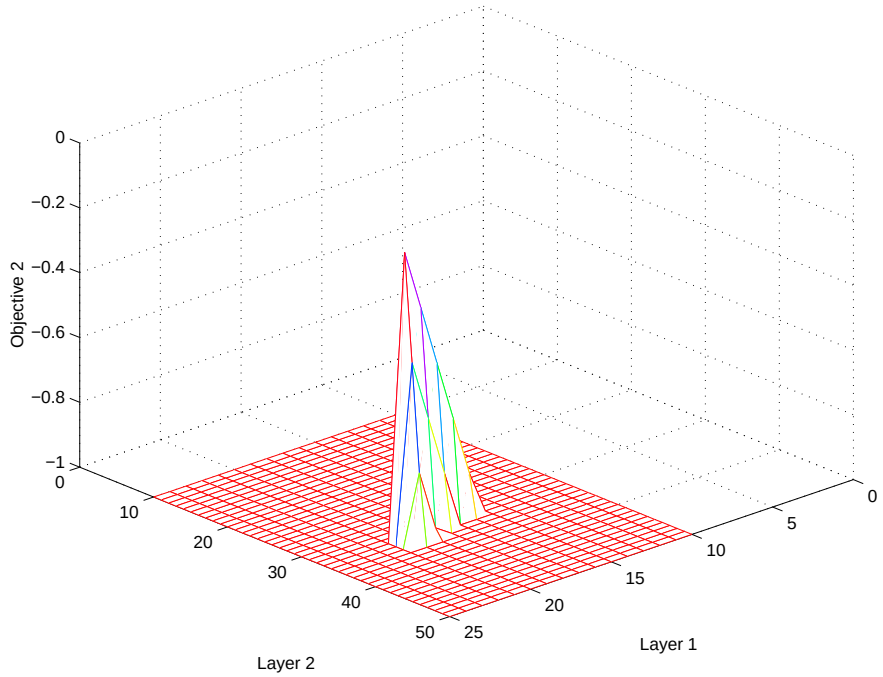

Figure 35 Fitness Landscape Showing Objective Value 2 with Thicknesses for Layers 1 and 2 at 5.0 THz

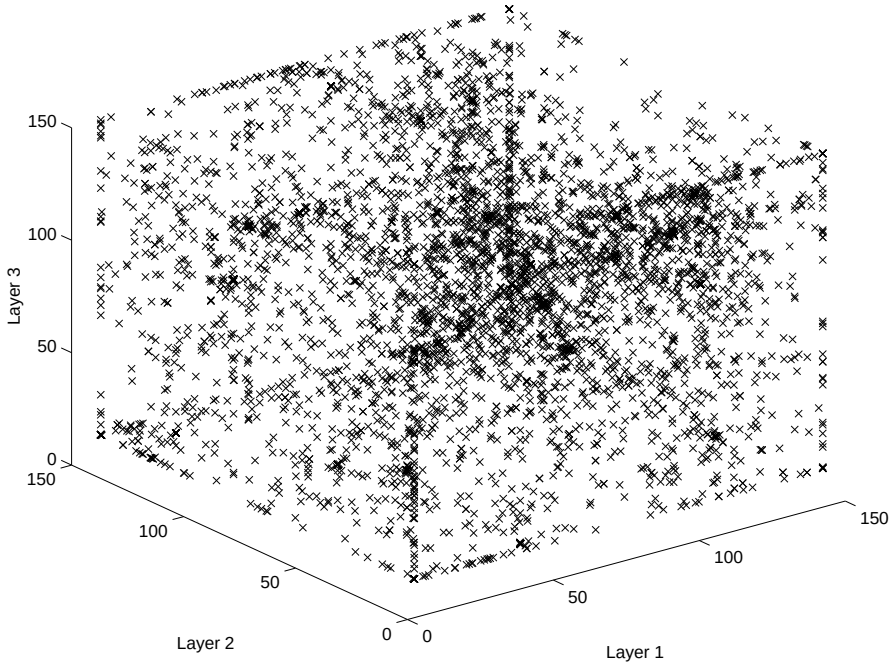

Figure 36 Values Searched for Layers 1,2 and 3 at 5.0 THz

In the early 1980s computer manufacturers were developing message passing libraries for their own multicomputer architectures. Because each of these companies used the same general send and receive format for communication the Message Passing Interface was developed to provide users with a standard for writing message passing programs. Presented to the computing community in November 1993 and adopted as an industry standard MPI is, in essence, an interface specification that was designed to be practical, portable, efficient and flexible [10].

MPI is in widespread use today and supported on most high performance computing (HPC) platforms. Over one hundred and fifteen routines are specified, but the novice programmer can build a routine with the six MPI commands described in Table 11

Table 11 **Six Common MPI Functions**

Name	Functionality
MPI_INIT()	Initializes MPI program
MPI_COMM_SIZE()	Returns # of cooperating processes
MPI_COMM_RANK()	Returns process identifier
MPI_SEND()	Sends a message
MPI_RECV()	Receives a message
MPI_FINALIZE()	Terminates MPI

The Aspen Beowulf cluster is a multicomputer architecture with forty-eight nodes and two processors per node connected to a fast-ethernet backplane. The interconnection network is configured with a crossbar switch, which conveniently makes the network completely connected. The cluster utilizes a distributed memory architecture that can be accessed at 100 Mb/s through the use of some message passing protocol. These details are contained in Table 12.

Table 12 **Aspen Configuration**

Nodes	48
Processors/Node	2/Node
Backplane	Fast-ethernet
Interconnection	Crossbar Switch
Memory	Distributed
Disk I/O	Raid 5
RAM	1 GByte
Cache L1, L2	16, 256 KB

The following sections, adapted from [47], describe metrics used to evaluate the performance of MOEAs.

H.1 Error Ratio:

The Error Ratio (ER) metric reports the number of vectors in PF_{known} that are not members of PF_{true}. This metric requires that the researcher knows PF_{true}. The mathematical representation of this metric is shown in equation 37:

$$ER \triangleq \frac{\sum_{i=1}^{n} e_i}{n} \tag{37}$$

where n is the number of vectors in PF_{known} and e_i is a zero when the i vector is an element of PF_{true} or a 1 if i is not an element. [18]

So when $ER = 0$, the PF_{known} is the same as PF_{true}; but when $ER = 1$, this indicates that none of the points in PF_{known} are in PF_{true}.

H.2 Two Set Coverage:

The Two Set Coverage (CS) metric is named in [18], but was originally defined in [89]. This metric compares the coverage of two competing sets and outputs the percentage of individuals in one set dominated by the individuals of the other set. This metric does not require that the researcher have knowledge of PF_{true}. The equation for this metric is shown in equation 38:

$$CS(X', X'') \triangleq \frac{|a'' \in X''; \forall a' \in X' : a' \succeq a''|}{|X''|} \tag{38}$$

where $X', X'' \subseteq X$ are two sets of phenotype decision vectors, and (X', X'') are mapped to the interval $[0, 1]$. This means that $CS = 1$ when X' dominates or equals X''.

Zitzler uses this a metric in several publications [91, 92, 94].

H.3 Generational Distance:

The Generational Distance (GD) metric is defined in [18, 81]. It reports how far, on average, PF_{known} is from PF_{true}. This metric requires that the researcher knows PF_{true}. It is mathematically defined in equation

$$GD \triangleq \frac{(\sum_{i=1}^{n} d_i^p)^{1/p}}{n} \tag{39}$$

where n is the number of vectors in PF_{known}, $p = 2$, and D_i is the Euclidean distance between each member and the closest member of PF_{true}, in the phenotype space. When $GD = 0$, $PF_{known} = PF_{true}$.

H.4 Hyperarea and Ratio:

The Hyperarea (H) and Ratio (HR) metrics, discussed in [18, 92], define the area of coverage that PF_{known} has with respect to the objective space. This would equate to the summation of all the areas of rectangles, bounded by the origin and $(f_1(\vec{x}), f_2(\vec{x}))$, for a two-objective MOEA. Mathematically, this is described in equation 40:

$$H \triangleq \left\{ \bigcup_i a_i | v_i \in PF_{known} \right\} \tag{40}$$

where v_i is a nondominated vector in PF_{known} and a_i is the area of the calculated between the origin and vector v_i. But if PF_{known} is not convex, the results can be misleading. It is also assumed in this model that the origin is $(0, 0)$

The hyperarea ratio metric definition can be seen in equation 41:

$$HR \triangleq \frac{H_1}{H_2} \tag{41}$$

where H_1 is the PF_{known} hyperarea and H_2 is the hyperarea of PF_{true}. This results in $HR \geq 1$ for minimization problems and $HR \leq 1$ for maximization problems. For either type of problem, $PF_{known} = PF_{true}$ when $HR = 1$. This metric requires that the researcher knows PF_{true}.

H.5 Spacing:

The Spacing (S) metric outputs the spread of the vectors in PF_{known}. Coello describes this metric from [**?**] in his book [18]. This metric measures the distance variance of neighboring vectors in PF_{known}. Equation 42 defines this metric.

$$S \triangleq \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \tag{42}$$

and

$$d_i = min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|) \tag{43}$$

where $i, j = 1 \ldots, n$, $\bar{d}$ is the mean of all d_i, and n is the number of vectors in PF_{known}. When $S = 0$, all members are spaced evenly apart. This metric does not require the researcher to know PF_{true}.

H.6 Overall Nondominated Vector Generation Ratio:

The Overall Nondominated Vector Generation Ratio (ONVGR) metric measures the total number of nondominated vectors during MOEA execution and divides it by the number of vectors found in PF_{true}. Coello [18] defines this metric as shown in equation 44:

$$ONVG \triangleq \frac{PF_{false}}{PF_{true}} \tag{44}$$

When $ONVGR = 1$ this states only that the same number of points have been found in both PF_{true} and PF_{known}. It does not infer that $PF_{true} = PF_{known}$. This metric requires that the researcher knows PF_{true}.

H.7 Progress Measure:

For single-objective EAs, Bäch [5] defines a metric that measures convergence velocity. The Progress Measure (RP) single-objective metric is applied to multiobjective MOEAs in [18], and is shown in equation 45:

$$RP \triangleq ln\sqrt{\frac{G_1}{G_T}} \tag{45}$$

where G_1 is the generational distance for the first generation and G_T is the distance for generation T. Recall that generational distance was defined in equation 39 and it measures the average distance from PF_{true} to PF_{known}. This metric requires that the researcher knows PF_{true}.

H.8 Generational Nondominated Vector Generation:

The Generational Nondominated Vector Generation (GNVG) is a simple metric, defined in [18] that lists the number of nondominated vectors produced for each generation. This is defined in equation 46

$$GNVG \triangleq |PF_{current}(t)| \tag{46}$$

This metric does not require the researcher know PF_{true}.

H.9 Nondominated Vector Addition:

The Nondominated Vector Addition (NVA) metric, defined in [18], calculates the number of nondominated vectors gained or lost from the previous PF_{known} generation. Equation 47 defines this metric.

$$NVA \triangleq |PF_{known}(t)| - |PF_{known}(t-1)| \tag{47}$$

But this metric can be misleading when a new vector dominates two or more vectors from the previous generation. In addition, this metric may remain static over the course

of several generations while new points are added that dominate others from the previous generation. This metric does not require the researcher know PF_{true}.

Table 13 lists the various MOEA metrics and states whether they require PF_{true} or explicitly compare results from one generation to another.

Table 13 **Summary of MOEA Metrics**

Metric Name	PF_{true} required?	Generational Metric?
Error Ratio	Yes	No
Two Set coverage	No	No
Generational Distance	Yes	Yes
Hyperarea	No	No
Hyperarea Ratio	Yes	No
Spacing	No	No
ONVGR	Yes	Yes
Progress Measure	Yes	Yes
GNVG	No	Yes
Nondominated Vector Addition	No	Yes

Bibliography

1. Aarts, E. and Lentra, J. K. *Local Search in Combinatorial Optimization*. John Wiley and Sons Ltd., 1997.
2. Adamidis, P. *Review of Parallel Genetic Algorithms Bibliography*. Technical Report, Thessaloniki, Greece: Aristotle University of Thessaloniki, November 1994.
3. Babbar, M. and Minsker, B. *A Multiscale Master-Slave Parallel Genetic Algorithm wiht Application to Groundwater Remediation Design*. University of Illinois, Urbana, IL, 2002.
4. Back, T. *Evoluationary Algorithms in Theory and Practice*. Oxford Univerity Press, 1996.
5. Bäck, T. A. *Evolutionary Algorithms in Theory and Practice*. New York - Oxford: Oxford University Press, 1996.
6. Barnard, S. *PMRSB: Parallel Multilevel Recurisve Spectral Bisection*. Technical Report, Sunnyvale, CA: Cray Research, Inc., 1995.
7. Beck, M., Hofstetter, D., Aellen, T., and Faist, J. *Continuous Wave Operation of Quantum Cascade Lasers Emitting Near 5.6 um*. Technical Report, Zurich, Switzerland: University of Neuchatel, 2002.
8. Bennage, W. and Dhingra, A. K. "Singel and Multiobjective Structural Optimization in Discrete-Continuous Variables Using Simulated Annealing," *International Journal for Numerical Methods in Engineering*, *38*:2753–2773 (1995).
9. Botvidzon, J. and Wessling, A. "Quantum Cascade Laser," (February 2001). Project in Optoelectronics.
10. Bramley, R., "Basic MPI," 2003. http://www.cs.indian.edu/classes/b673/notes/mpi1.html.
11. Cantu-Paz, E. "A Survey of Parallel Genetic Algorithms," (May 1997). IlliGAL Report No. 97003.
12. Capasso, F., Gmachl, C., Sivco, D. L., and Cho, A. Y. "Quantum Cascade Lasers," *Physics Today* (May 2002).
13. Capasso, F., Paiella, R., Martini, R., Colombelli, R., Gmachl, C., Myers, T. L., Taubman, M. S., Williams, R. M., Bethea, C. G., Unterrainer, K., Hwang, H. Y., Sivco, D. L., Cho, A. Y., Sergent, A. M., Liu, H. C., and Whittacker, E. A. "Quantum Cascade Lasers: Ultrahigh-Speed Operation, Optical Wireless Communicatwion, Narrow Linewidth, and Far-Infrared Emission," *Journal of Quantum Electronics*, *38*(6) (June 2002).
14. Chattopadhyay, A. and Seeley, C. E. "A Simulated Annealing Technique for Multiobjective Optimization of Intelligent Structures," *Smart Motor Structures*, 98–106 (February 1994). Department of Mechanical and Aerospace Engineering, Arizona State University.

15. Chow, W. W. and Koch, S. W. *Semiconductor-Laser Fundamentals*. New York: Springer-Verlag Heidelberg, 1999.

16. Chow, W. W., Koch, S. W., and Sargent, M. *Semiconductor Laser Physics*. Berlin: Springer-Verlag, 1994.

17. Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B. *Evoluationary Algorithms for Solving Multi-objective Problems*. New York: Kluwer Academic Press, 2002.

18. Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic Publishers, May 2002.

19. Cohoon, J., Martin, W., and Richards, D. "A Multi-population Genetic Algorithm for Solving the K-partition Problem on Hypercubes," *Proceedings of the Fourth International Conference on Genetic Algorithms*, 244–248

20. Collette, Y. *Multiobjective Optimization: Principles and Case Studies*. Holding Publisher, October 2003.

21. Dengiz, B., Altiparmak, F., and Smith, A. E. "Local Search Genetic Algorithm for Optimization of Highly Reliable Communications Networks.".

22. Eiben, A. E. *Evolutionary exploration of the search spaces*, 178–188. Springer-Verlag, 1996.

23. Esaki, L., Chang, L. L., and Tsu, R. "A One-dimensional 'Superlattice' in Semiconductors." *Proceedings 12th International Conference on Low Temperature Physics*. 551. 1970.

24. Eshelman, L. J., Caruana, R. A., and Schaffer, J. D. "Biases in the crossover landscape." *Proceedings of the 3rd International Conference on Genetic Algorithms*, edited by J. D. Schaffer. 10–19. Morgan Kaufmann, 1989.

25. Esslinger, M. *An Artificial Immune System Strategy for Robust Chemical Spectra Classification via Distributed Heterogeneous Sensors*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, 2003.

26. Fleming, P. J. and Fonseca, C. M. "Genetic Algorihtms for Multi-objective Optimization: Formulation, Discussion and Generalization." *5th International Congress on Genetic Algorithms (ICGA)*. 416–423. 1993.

27. Fonseca, C. M. and Fleming, P. J. "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization." *Proceedings of the Fifth International Conference on Genetic Algorithms*, edited by Stephanie Forrest. 416–423. San Mateo, CA: Morgan Kauffman Publishers.

28. Fukuda, M. *Optical Semiconductor Devices*. New York: John Wiley and Sons, Inc., 1999.

29. Gagnon, J. *Undergraduate Thesis*. MS thesis, Worchester Polytechnic Institute, 2002.

30. Garrett, G., Huang, J., Goltz, M., and Lamont, G. B. "Parallel Real-Valued Genetic Algorihtms for Bioremediation Optimization of TCE-Contaminated Groundwater," *IEEE Congress on Evoluationary Computation (CEC)*, 2183–2189 (1999).

31. Gendreau, M. "An Introduction to Tabu Search." University of Montreal, July 2002.

32. Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley Publishing, 1989.

33. Golub, M. and Budin, L. "An Asynchronous Model of Global Parallel Genetic Algorithms." *2nd ICSC Symposium on Engineering of Intelligent Systems EIS2000*. 353–359. 2000.

34. Golub, M. and Jakobovic, D. "A New Model of Global Parallel Genetic Algorithm." University of Zagreb, Croatia.

35. Gordon, V. and Whitley, D. "Serial and Parallel Genetic Algorithms as Function Optimizers," 177–183 (1993).

36. Griffiths, D. J. and Harris, E. G. *Introduction to Quantum Mechanics*. Englewood Cliffs, NJ: Prentice Hall, 1995.

37. Hao, J. and Pannier, J. "Simulated Annealing and Tabu Search for Constraint Solving," *Fifth Intl. Symposium on Artificial Intelligence and Mathematics* (January 1998).

38. Holland, J. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

39. Holland, J. and Goldberg, D. E. *Introduction to Genetic Algorithms with Java Applets*. http://cs.felk.cvut.cz/ xobitko/ga.

40. Holonyak, N. J., Kolbas, R. M., Dupuis, R. D., and Dapkus, P. D. "Quantum-Well Heterostructure Lasers," *IEEE Journal of Quantum Electronics*, *QE-16*(2):170–186 (1980).

41. Horn, J. and N., N. *Multiobjective Optimization using the Niched Pareto Genetic Algorithm*. Technical Report 93005, Urabana, IL: University of Illinois at Urbana-Champaign, 1993.

42. Horn, J., Nafpliotis, N., and Goldberg, D. E. "A Niched Pareto Genetic Algoirthm for Multiobjective Optimization," 82–87 (1994).

43. Huebner, K. H., Dewhirst, D. L., Smith, D. E., and Byrom, T. G. *The Finite Element Method for Engineers*. John Wiley and Sons, Inc., 2001.

44. Jin, J. *The Finite Element Method in Electromagnetics*. John Wiley and Sons, Inc., 1993.

45. K., D. and Goldberg, D. E. *mGA in C: A Messy genetic Algorithm in C*. Technical Report 91008, Illinois Genetic Algorihtms Laboratory (IlliGAL), September 1991.

46. Khan, N. *Bayesian Optimization Algorithms for Multiobjective and Hierarchically Difficult Problems*. MS thesis, University of Illinois at Urbana-Champaign, Urbana, IL, July 2003.

47. Kleeman, M. *Optimization of Heterogeneous UAV Communications using the Multiobjective Quadratic Assignment Problem*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, 2004.

48. Knarr, M. R., Goltz, M. N., Lamont, G. B., and Huang, J. "In-situ Bioremediation of Perchlorate-Contaminated Groundwater using Multi-Objective Parallel Evoluationary Alorithm," (2002).

49. Knowles, J. and Corne, D. "M-PAES: A Memetic Algorithm for Multiobjective Optimization." *2000 Comgress on Evolutionary Computation*. 325–332. July 2000.

50. Kuehl, B. and Robert, O. *Design of Experiments: Statistical Principles of Research Design and Analysis*. Duxbury Press, 2000.

51. Kumar, V., Gupta, A., Grama, A., and Karypis, G. *Introduction to Parallel Computing*. Addison-Wesley, 2003.

52. Kumar, V., Gupta, A., Grama, A., and Karypis, G. *Introduction to Parallel Computing*. Addison-Wesley, 2003.

53. Lin, S.C., P. W. I. and Goodman, E. "Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach," *IEEE* (1994). Michigan State University.

54. Marciniak, M. A. *Optical Characterization of Indium Arsenide Antimonide Semiconductors Grown by Molecular Beam Epitaxy*. PhD dissertation, Air Force Institute of Technology, Wright Patterson AFB, OH, September 1995.

55. Menon, V. "Design, Fabrication and Characterization of Quantum Cascade Terahertz Emitters," (2001). PhD Dissertation.

56. Menon, V. M., Goodhue, W. D., Karakashian, A. S., and Ram-Mohan, L. R. "Waveguide Design Optimization for a Quantum Cascade Laser Emitting at 77 um," *Physica E* (2000).

57. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolutionary Programs*. Springer, 1996.

58. Milton, J. S. and Arnold, J. C. *Introduction to Probability and Statistics*. McGraw-Hill Primis Custom Publishing, 2002.

59. Mounce, G. D. *Building Blocks for Time-Resolved Laser Emission in Mid-Infrared Quantum Well Lasers*. MS thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, 2001.

60. Naudts, B. and Schippers, A., "A Motivated Definition of Exploitation and Exploration," February 1999. University of Antwerp, Belgium.

61. Obitko, M. "Introduction to Genetic Algorithms," (1998). http://cs.felk.cvut.cz/ xobitko/ga/encoding.html.

62. Osyczka, A. *Multicriteria Optimization for Engineering Design*.

63. Otten, C. J., "For Quantum Confinement, Size Matters, but so does Shape," October 2003. Washinton University, St. Louis.

64. Pelikan, M. and Goldberg, D. E. "Hierarchical Problem Solving and the Bayesian Optimizatoin Algorithm." *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*. 267–274. 2000.

65. Pierret, R. F. *Semiconductor Device Fundamentals*. Addison-Wesley Publishing Company, 1996.

66. Priolo, F. and Fabrizio, E. D. *Molecular Beam Epitaxy*, 2004. http://www.elettra.trieste.it/experiments/beamlines/lilit/htdocs/people/luca/tesihtml/node24.html.

67. R, and Digest, D. "Exploring the Capabilities of Terahertz Imaging," *Medical Device and Diagnostic Industry* (February 2002).

68. Ram-Mohan, L. R., "Quantum Cascade Terahertz Lasers," 2003. Air Force Research Labs, Wright Patterson AFB, OH.

69. Rieck, H. *Semiconductor Lasers*. London: MacDonald, 1970.

70. Schneckenburger, T. and Huber, M. "Heterogenous Partitioning in a Workstation Network." *8th International Parallel Processing Symposium*. 72–77. April 1994.

71. Schombert, J. *Bohr Atom*. University of Oregon, Eugene, OR, 2004. http://zebu.uoregon.edu/ ∼js/glossary/bohr_atom.html.

72. Spiegel, M. R. and Stephens, L. J. *Statistics*. New York: McGraw-Hill, 1999.

73. Srinivas, N. and Deb, K. *Evolutionary Computation 2*, chapter 3, 221–248. Fall 1994.

74. Stracuzzi, D. "Some Methods for the Parallelization of Genetic Algorithms," (May 1998).

75. Tan, K. C., Khor, E. F., Lee, T. H., and Yang, Y. J. "A Tabu-Based Exploratory Evoluationary Algorithm for Multiobjective Optimization," *Artificial Intelligence Review*, *19*:231–260 (May 2003).

76. Tanese, R. "Distributed Genetic Algorithms," *Proceedings of the Third International Conference on Genetic Algorithms*, 434–439 (2001).

77. Tihov, M. I. "Chemical Sensors Based on Distributed Feedback Quantum Cascade Laser for Environmental Monitoring," (April 2003).

78. Tomaich, T. "A Genuinely Multi-Dimensional Upwinding Algorithm for the Navier-Stokes Equations on Unstructured Grids Using a Compact, Highly-Parallelizable Spatial Discretization," (1995).

79. Triki, E., Collette, Y., and Siarry, P. *Empirical Study of Simulated Annealing Aimed at Improved Multiobjective Optimization*. EDF Pole Industry, Clamart, France, 2000.

80. Van Veldhuizen, D. A. *Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations*. PhD dissertation, Air Force Institute of Technology, Wright Patterson AFB, OH, 1999.

81. Van Veldhuizen, D. A. and Lamont, G. B. "Evolutionary Computation and Convergence to a Pareto Front." *Late Breaking Papers at the Genetic Programming 1998*

Conference, edited by John R. Koza. 221–228. Stanford University, California: Stanford University Bookstore, July 1998.

82. Van Veldhuizen, D. A. and Lamont, G. B., "On Measuring Multiobjective Evolutionary Algorithm Performance," 2002. Air Force Institute of Technology, Wright Patterson AFB, OH.

83. Van Veldhuizen, D. A., Zydallis, J. B., and Lamont, G. B. "Considerations in Engineering Multiobjective Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, *Vol 7* (April 2003).

84. Van Zeghbroeck, B., "Principles of Semiconductor Devices," 2002. http://ece-www.colorado.edu/ bart/book/book.

85. Wacher, A. "Gain in Quantum Cascade Lasers and Superlattices: A Quantum Transport Theory," *Physical Review* (June 2002).

86. Wales, J. and Sanger, L. *Stimulated Emission*. Wikipedia Foundation, Inc., May 2004. http://en.wikipedia.org/wiki/Stimulated_emission.

87. Watson, J. *Photon Emission and Absorption*. University of Aberdeen, Aberdeen, UK. http://vcs.abdin.ac.uk/ENGINEERING/lasers/absorption.html.

88. Zaki, M., L. W. and S., P. "Customized Dynamic Load Balancing for a Network of Workstations," *Proceedings of HPDC* (1996).

89. Zitzler, E., Deb, K., and Thiele, L. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, *8*(2):173–195 (Summer 2000).

90. Zitzler, E., Laumanns, M., and Thiele, L. "SPEA2: Improving the Strength Pareto Evoluationary Algorithm." *Evoluationary Methods for Deisgn, Optimization and Control with Applications to Industrial Problems*, edited by et. al. Giannakoglou, K. September 2001.

91. Zitzler, E. and Thiele, L. *An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach*. Technical Report 43, Zurich, Switzerland: Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), May 1998.

92. Zitzler, E. and Thiele, L. "Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study." *Parallel Problem Solving from Nature V*, edited by A. E. Eiben. 292–301. Amsterdam: Springer-Verlag, September 1998.

93. Zitzler, E. and Thiele, L. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *IEEE Transactions on Evolutionary Computation, 34*. 257–271. November 1999.

94. Zitzler, E. and Thiele, L. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, *3*(4):257–271 (November 1999).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
15-06-2004	**Master's Thesis**	Aug 2003 – Jun 2004

4. TITLE AND SUBTITLE

Optimization of a Quantum Cascade Laser Operating in the Terahertz Frequency Range Using a Multiobjective Evolutionary Algorithm

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Keller, Traci A., 1st Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCE/ENG/04-03

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/SN
Attn: Thomas R. Nelson
2241 Avionics Circle, Bldg 2 — DSN: 785-1874, ext. 3361
WPAFB, OH 45433 — e-mail: Thomas.Nelson@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

A quantum cascade (QC) laser is a specific type of semiconductor laser that operates through principles of quantum mechanics. In less than a decade QC lasers are already able to outperform previously designed double heterostructure semiconductor lasers. Because there is a genuine lack of compact and coherent devices which can operate in the far-infrared region the motivation exists for designing a terahertz QC laser. A device operating at this frequency is expected to be more efficient and cost effective than currently existing devices. It has potential applications in the fields of spectroscopy, astronomy, medicine and free-space communication as well as applications to near-space radar and chemical/biological detection.

The overarching goal of this research was to find QC laser parameter combinations which can be used to fabricate viable structures. To ensure operation in the THz region the device must conform to the extremely small energy level spacing range from ~10-15 meV. The time and expense of the design and production process is prohibitive, so an alternative to fabrication was necessary. To accomplish this goal a model of a QC laser, developed at Worchester Polytechnic Institute with sponsorship from the Air Force Research Laboratory Sensors Directorate, and the General Multiobjective Parallel Genetic Algorithm (GenMOP), developed at the Air Force Institute of Technology, were integrated to form a computer simulation which stochastically searches for feasible solutions.

15. SUBJECT TERMS

Quantum cascade laser, multiobjective optimization, multiobjective evolutionary algorithm and terahertz.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Gary B. Lamont, AD-24, DAF
U	U	U	UU	125	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext 4718; e-mail: gary.lamont@afit.edu